# Swarm Intelligence and Computational Intelligence Optimization Based Blind Equalization Algorithm

郭业才 著

# 群体智能与计算智能优化的盲均衡算法

清华大学出版社

# Swarm intelligence and Computational Intelligence Optimization Based Blind Equalization Algorithm

# 群体智能与计算智能优化的盲均衡算法

郭业才　著

清华大学出版社
北　京

## 内 容 提 要

本书针对盲均衡理论与算法研究中初始权向量优化的难题，以智能群算法和智能计算理论为工具，开展了盲均衡算法性能优化的研究。主要内容有：基于遗传算法（含自适应遗传算法、模拟退火遗传算法和改进混合遗传算法）优化的正交小波类盲均衡算法；基于混沌算法或混沌支持向量机算法优化的正交小波加权多模盲均衡算法；基于免疫克隆算法优化的正交小波盲均衡算法及正交小波支持向量机盲均衡算法；基于粒子群算法（免疫克隆粒子群算法、量子粒子群算法及动态粒子群算法）优化的正交小波类盲均衡算法；基于人工鱼群算法（模拟退火与人工鱼群变异混合算法、混沌人工鱼群算法、免疫人工鱼群算法、量子人工鱼群算法）优化的多模盲均衡算法；基于DNA遗传算法（禁忌搜索自适应双链DNA遗传算法、多种群禁忌搜索DNA遗传算法）优化的多模盲均衡算法；基于DNA智能群算法（DNA人工鱼群算法、DNA遗传蛙跳算法、DNA遗传蝙蝠算法）优化的多模盲均衡算法。

全书结构分明，各章自成闭环系统、层层递进；内容既相对独立又相互联系，集系统性与新颖性于一体，是盲均衡算法智能优化研究成果的集中体现。本书适合于信息与通信工程、水声工程、控制科学与工程、智能科学与技术等学科专业的研究生和科研人员阅读。



**图书在版编目（CIP）数据**

群体智能与计算智能优化的盲均衡算法/郭业才著．—北京：清华大学出版社，2018
ISBN 978-7-302-50404-7

Ⅰ．①群…　Ⅱ．①郭…　Ⅲ．①盲均衡算法　Ⅳ．①TN911.1

中国版本图书馆CIP数据核字（2018）第122992号

**责任编辑**：梁　颖
**封面设计**：常雪影
**责任校对**：李建庄
**责任印制**：李红英

**出版发行**：清华大学出版社
**网　　址**：http://www.tup.com.cn，http://www.wqbook.com
**地　　址**：北京清华大学学研大厦A座　　**邮　　编**：100084
**社 总 机**：010-62770175　　**邮　　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质量反馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**课件下载**：http://www.tup.com.cn，010-62795954
**印 装 者**：三河市铭诚印务有限公司
**经　　销**：全国新华书店
**开　　本**：155mm×235mm　　**印　　张**：24　　**字　　数**：346千字
**版　　次**：2018年8月第1版　　**印　　次**：2018年8月第1次印刷
**定　　价**：128.00元

产品编号：078378-01

# 前言

信道均衡技术可有效消除码间干扰和噪声、减小误码率、提高通信质量，可分为时域均衡和频域均衡、自适应均衡（简称均衡，需要训练序列）和盲自适应均衡（简称盲均衡，不需要训练序列）。其中，时域盲均衡技术是基于 1975 年日本学者 Y. Sato 等人提出的信息自恢复、信道盲辨识和信道盲均衡等新思想的基础上快速发展起来的，出现了很多有效和新颖的算法，主要用收敛速度、均方误差或码间干扰、信干噪比、计算复杂度等指标考察算法性能。盲均衡技术的本质就是设计与信道特性相反的逆波器（即盲均衡器），其核心是设计更新盲均衡器权向量的算法，而盲均衡算法设计涉及盲均衡性能准则（又称代价函数）、误差函数、步长、权向量初始化方法和数学工具等，这些因素直接影响算法的性能。

本书作者及其团队自 2002 年以来，基于水声信道、卫星信道、移动信道等通信环境，开展了盲均衡算法研究，取得了一些有益成果，并相继出版。2007 年由合肥工业大学出版社出版了国内第一部盲均衡研究专著《自适应盲均衡技术》，其内容主要以高阶累积量理论为工具，研究了变步长盲均衡算法、分数间隔盲均衡算法和双模式盲均衡算法等。2009 年由气象出版社出版了国内第二部盲均衡研究专著《通信信道盲均衡算法与仿真》，其内容主要是关于误差函数对盲均衡算法性能影响的研究成果。2011 年由科学出版社出版了专著《模糊小波神经网络盲均衡理论、算法与实现》，该书内容反映了以模糊数学、神经网络、高等数理统计和小波理论为工具的盲均衡算法研究成果。

本书内容是在前 3 部专著的基础上，重点汇集了以智能群算法和智能计算方法为工具开展正交小波盲均衡算法性能优化研究所取得的新进展。

主要内容为：

(1) 基于遗传算法优化的盲均衡算法；

(2) 基于混沌算法优化的盲均衡算法；

(3) 基于免疫克隆算法优化的盲均衡算法；

(4) 基于粒子群算法优化的盲均衡算法；

(5) 基于人工鱼群算法优化的盲均衡算法；

(6) 基于 DNA 遗传算法优化的盲均衡算法；

(7) 基于 DNA 智能群算法优化的盲均衡算法。

本书成果在研究过程中，得到了全国优秀博士学位论文作者专项资金(200753)、国家自然科学基金(61673222)、江苏省自然科学基金(BK2009410)、江苏省高校自然科学基金(08KJ B510010)、江苏省高校自然科学研究重大项目(13KJA510001)、安徽省自然科学基金(0504 20304)、安徽省高校自然科学研究项目(2003KJ092、KJ2010A096)、江苏高校优势学科建设工程二期"信息与通信工程"、江苏高校品牌专业建设工程一期(PPZY2015B134)等项目资助。其中的一些成果已经陆续在国内外重要学术会议和期刊上发表，同时也吸收了一些其他作者的成果，反映了盲均衡算法优化研究的最新动态和前沿趋势。

与本书作者共同研究的人员有：韩迎鸽、饶伟、杨超、廖娟、丁锐、季童莹、胡苓苓、黄伟、孙静、丁雪洁、王卫、徐文才、李宝鸽、孙凤、柏鹤、张冰龙、王惠、姚超然、陆璐等。全书在编写过程中，参阅并引用了一些国内外作者的相关论著。本书的出版还得到了清华大学出版社的大力支持，在此一并表示诚挚的谢意！

由于作者水平有限，书中难免存在不当之处，敬请读者批评指正！

作　者

2018 年 5 月

# 目录

# 第 1 章　盲均衡概述

**【内容导语】** 本章内容包括均衡系统模型、Bussgang 类盲均衡算法、分数间隔盲均衡系统和超指数迭代盲均衡算法等分析；在基于正交小波变换的自适应均衡算法中，阐述了均衡器的正交小波表示、正交小波自适应均衡算法原理与性能等。

## 1.1　均衡系统模型

### 1.1.1　均衡系统等效基带模型

均衡系统等效基带模型如图 1.1 所示。

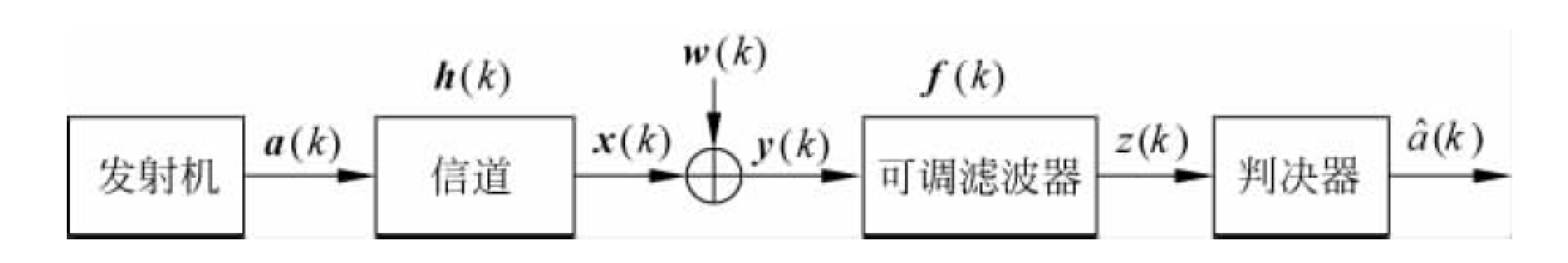



图 1.1　均衡系统等效模型

图 1.1 中,$\boldsymbol{a}(k)$表示信源信号向量,也即信道输入信号向量;$\boldsymbol{h}(k)$表示信道冲激响应向量;$\boldsymbol{w}(k)$表示加性信道噪声向量;$\boldsymbol{f}(k)$表示均衡器权向量或可调滤波器;$z(k)$表示均衡器输出。理论和实践证明,在数字通信系统中插入一种可调滤波器可以校正和补偿系统特性,减少码间干扰的影响。这种起补偿作用的滤波器称为逆滤波器或均衡器。由图可知,整个均衡系统的传输特性为将可调滤波器(或均衡器)设计成与信道特性匹配的滤波器,以补偿信道的畸变。也就是说,均衡器通常是用滤波器来实现的。使用滤波器来补偿失真的脉冲,判决器得到的解调输出样本,为经过均衡器修正过的或者清除了码间干扰之后的样本。

在图 1.1 所示的模型中,可调滤波器或均衡器常用结构主要有线性均衡器结构和判决反馈均衡器结构两种。其中,线性均衡器结构如图 1.2 所示。

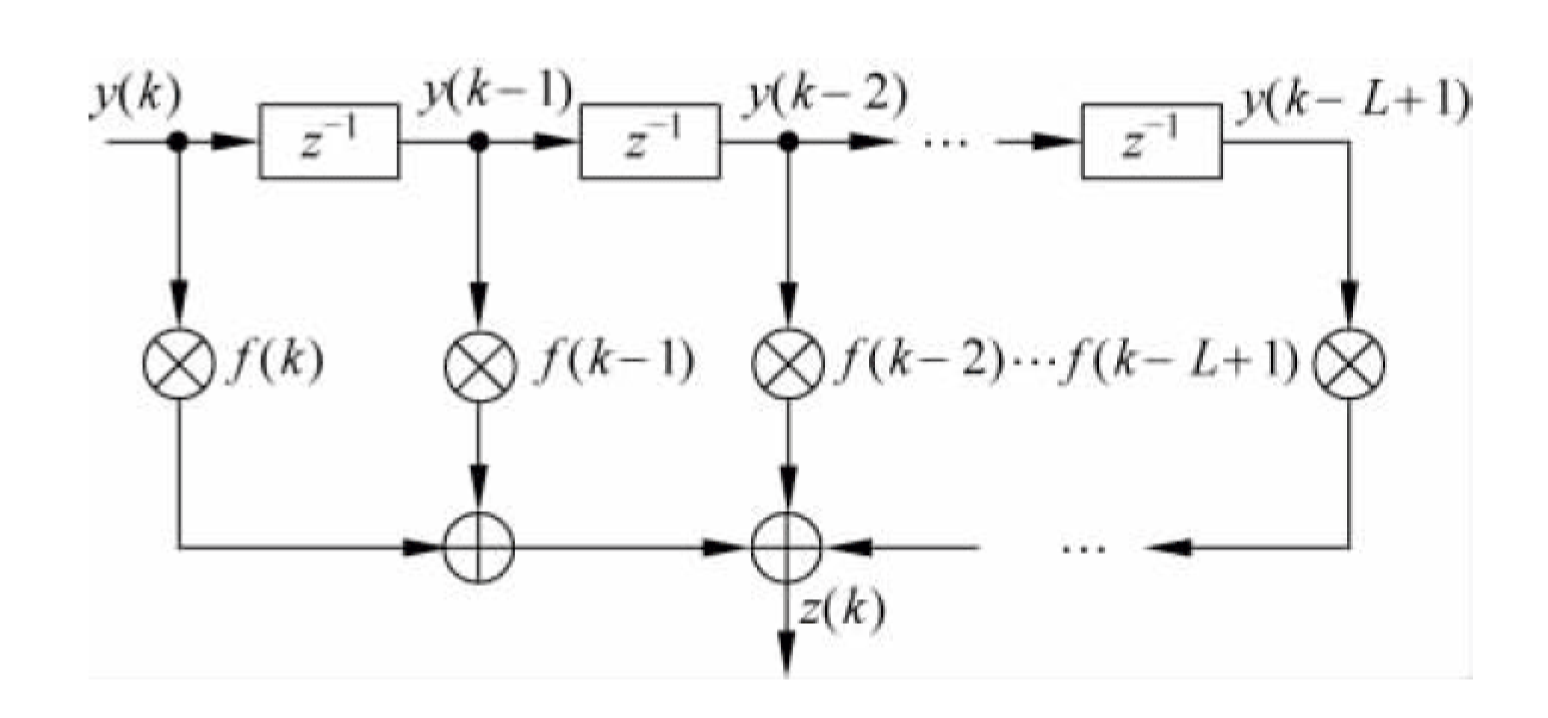



图 1.2　线性均衡器结构

图 1.2 中,均衡器输入向量 $\boldsymbol{y}(k)=\{y(k),y(k-1),\cdots,y(k-L+1)\}$,均衡器权向量为 $\boldsymbol{f}(k)=\{f(k),f(k-1),\cdots,f(k-L+1)\}$,均衡器输出为

$$z(k)=\sum_{i=0}^{L-1}f(i)y(k-i)=\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{1.1.1}$$

判决反馈均衡器的基本结构如图 1.3 所示。

图 1.3 中,判决反馈均衡器由一个前向滤波器和一个反馈滤波器构成,反馈滤波器以判决器的输出值为其输入值,然后将其输出值作为一路输入值输入前向滤波器,用来消除先前检测到的符号序列对当前符号序列所产生的干扰。

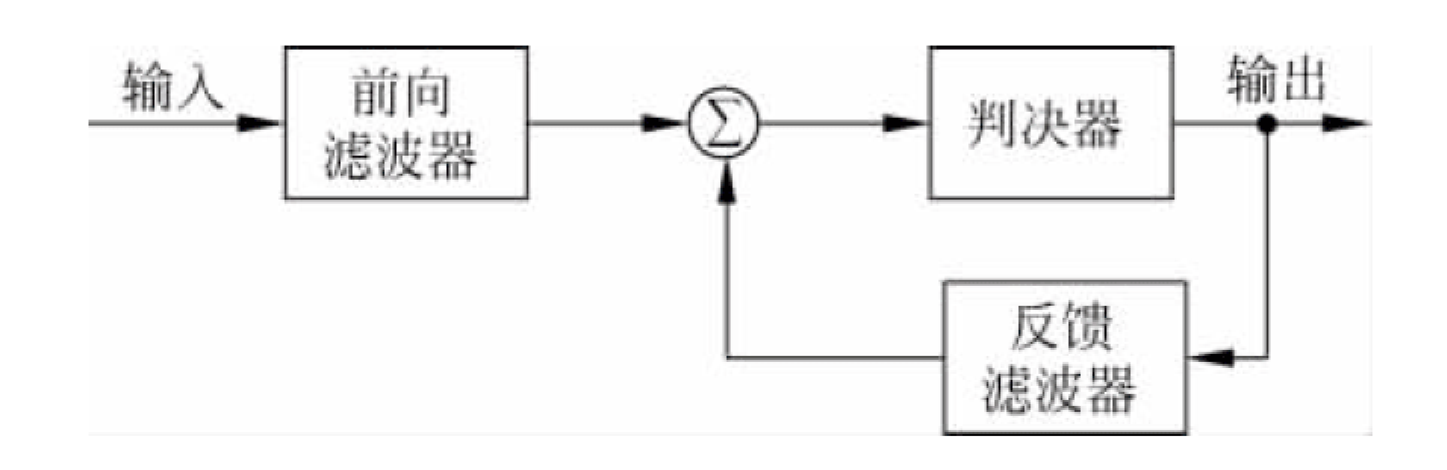



图 1.3　判决反馈均衡器的基本结构

在均衡系统中，均衡器权向量 $\boldsymbol{f}(k)$的最优向量 $\boldsymbol{f}_{opt}$如何获取呢？一种是用维纳滤波器；一种是采用自适应均衡器。而自适应均衡器又有传统自适应均衡器（非盲的）和盲自适应均衡器（简称盲均衡器）之分。现分述如下。

### 1.1.2　自适应均衡器与盲自适应均衡器

#### 1. 自适应均衡器

自适应均衡器直接从传输的实际数字信号中根据某种算法不断调整增益，因而能适应信道的随机变化，使均衡器总是保持最佳的工作状态，从而有更好的失真补偿性能。自适应均衡器一般包含两种工作模式，即训练模式和跟踪模式。首先，发射机发射一个已知的、定长的训练序列，以便均衡器可以做出正确的设置。典型的训练序列是一个二进制伪随机信号或是一串预先指定的数据位，而紧跟在训练序列后被传送的是用户数据，均衡器将通过递归算法来评估信道特性，并且修正滤波器权向量以对信道作出补偿。在设计训练序列时，要求做到即使在最差的信道条件下，均衡器也能通过这个训练序列获得最优的权向量，这样就可以使均衡器在收到训练序列后，其滤波系数已经接近于最佳值。而在接收数据时，均衡器的自适应算法就可以跟踪不断变化的信道，不断改变其滤波特性。均衡器从调整参数至形成收敛，整个过程是均衡器算法、结构和通信变化率的函数。为了能有效地消除码间干扰，均衡器需要周期性地做重复训练。在数字通信系统中用户数据是被分为若干段并被放在相应的时间段中传送的，每当收到新的时间段，

均衡器将用同样的训练序列进行修正。均衡器一般被放在接收机的基带或中频部分实现，基带包络的复数表达式可以描述带通信号波形，所以信道响应、解调信号和自适应算法通常都可以在基带部分被仿真和实现。

2. 盲自适应均衡器

盲自适应均衡器不需要训练序列就能补偿信道畸变所造成的影响，它是这样一类自适应算法：不需要外部提供期望响应，就能产生与期望恢复的输入信号在某种准则下最逼近的滤波器输出。换言之，算法对期望响应是“盲”的。然而，算法本身在自适应过程中，通过一个非线性变换产生期望响应的估计。这种自适应均衡器习惯称为盲自适应均衡器，简称盲均衡器。由盲均衡器的含义和图 1.1 可见，盲均衡的目的就是在已知信道输出信号 $\boldsymbol{y}(k)$ 的情况下，采用盲均衡算法调整均衡器权向量，使均衡器输出信号 $z(k)$ 在某种意义下是对 $a(k)$ 的最优估计，也就是 $z(k)$ 和 $a(k)$ 之间的偏差非常小，$z(k)$ 经过判决器的输出 $\hat{a}(k)$ 就基本上无失真地恢复了信源信号 $a(k)$。

在盲均衡算法中，当信源信号 $a(k)$ 为实信号且功率谱密度是常数，属于有限符号集，具有平稳独立同分布、零均值和单位方差特性，即：

$$E[a(k)]=0,\quad E[a(k)a(m)]=\begin{cases}1, & k=m\\ 0, & k\neq m\end{cases}\tag{1.1.2}$$

式中，$E[\cdot]$ 表示取平均值，即求数学期望。信源信号 $a(k)$ 的概率密度函数具有对称性，即

$$p_a(a(k))=\begin{cases}1/2\sqrt{3}, & -\sqrt{3}\leqslant a(k)<\sqrt{3}\\ 0, & \text{其他}\end{cases}\tag{1.1.3}$$

当信源信号 $a(k)$ 为复信号时，盲均衡算法要求信源信号的同相和正交两个分量必须同时满足上述条件。

$\boldsymbol{w}(k)$ 表示信道干扰噪声，为高斯白噪声；$\boldsymbol{h}(k)$ 表示信道的冲激响应且 $\boldsymbol{h}(k)=[h(k),h(k-1),\cdots,h(k-N+1)]^{\mathrm{T}}$（$[\cdot]^{\mathrm{T}}$ 表示转置运算）。

均衡器输入信号为

$$y(k) = \sum_{i=0}^{N-1} h_i(k)a(k-i) + w(k) \tag{1.1.4}$$

均衡器输入信号向量为

$$\boldsymbol{y}(k) = \boldsymbol{a}(k)\boldsymbol{h}(k) + \boldsymbol{w}(k) \tag{1.1.5}$$

均衡器输出信号为

$$z(k) = \boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) = \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k) = \sum_{i=-L}^{L} f_i(k)y(k-i) \tag{1.1.6}$$

式中,上标 T 表示转置。

## 1.2 Bussgang 类盲均衡算法

传统的自适应均衡技术需要外部供给期望信号,即在正式通信之前发射端将发送一段接收端已知的训练序列,通过自适应算法调节均衡器权向量,并最终达到反卷积目的。但是,训练序列的使用不但需要占用大量的信道带宽,而且在载波恢复的过程中一旦中断将直接导致均衡的失败,解决这一问题的有效途径是采用盲反卷积技术(盲均衡)。根据在何处对数据加非线性变换,盲均衡分为三大类。

(1) Bussgang 算法[1-3]:非线性的无记忆变换函数存在于自适应均衡器的输出端。

(2) 高阶或循环统计量算法:非线性变换存在于自适应均衡器的输入端。

(3) 非线性均衡算法:非线性存在于均衡器的内部,即使用非线性滤波器(Voltera 滤波器)或神经网络。

这里主要介绍 Bussgang 算法。

### 1.2.1 实基带信道的 Bussgang 算法

Bussgang 盲均衡算法结构如图 1.4 所示。

为了避免使用训练序列,Bussgang 盲均衡算法采用非线性估计器

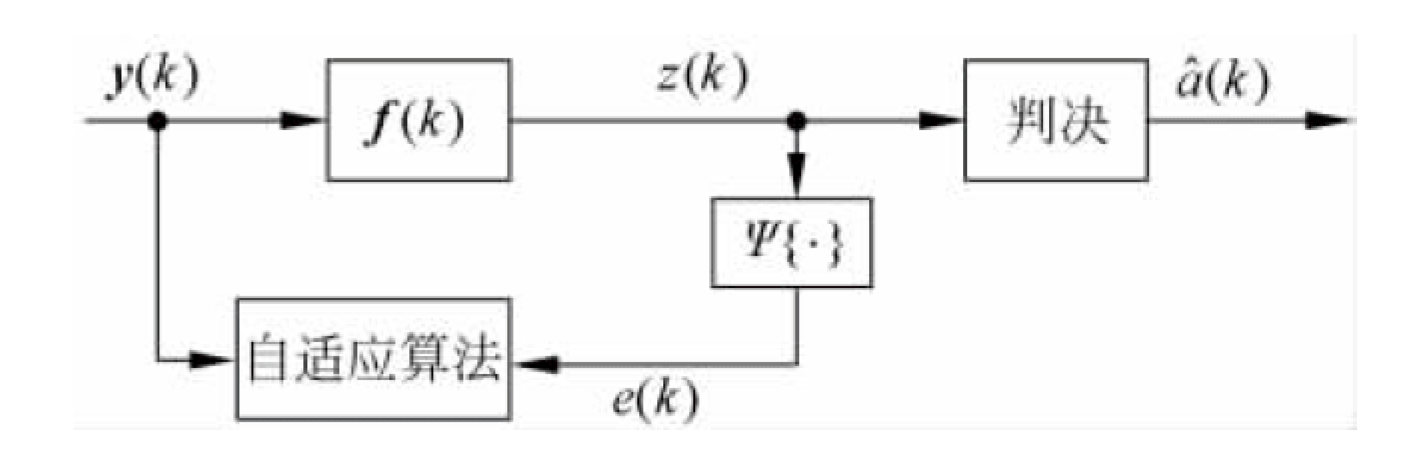



图 1.4　Bussgang 盲均衡结构

$g\{\cdot\}$对均衡器输出信号 $z(k)$进行非线性变换，以获得输入信号 $a(k)$的估计值$\hat{a}(k)$。$e(k)$是由误差生成函数 $\Psi\{\cdot\}$生成的误差信号，通过随机梯度算法对均衡器权向量 $\boldsymbol{f}(k)$进行调整。因此，Bussgang 盲均衡算法权系数更新公式为

$$f_i(k+1) = f_i(k) + \mu e(k) y(k-i) \tag{1.2.1}$$

式中，$f_i(k)$表示理想逆滤波器的权系数，它与信道冲激响应序列$\{h(k)\}$之间满足理想逆关系，即

$$\boldsymbol{f}(k) \otimes \boldsymbol{h}(k) = \delta(k), \quad \forall k \tag{1.2.2}$$

式中，$\otimes$表示卷积操作。现在用$\{f(k)\}$对均衡器输入信号 $y(k)$进行滤波，即

$$\begin{aligned}\sum_{i=-\infty}^{\infty} f_i(k) y(k-i) &= \sum_{i=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} f_i(k) h_m(k) a(k-i-m) \\ &= \sum_{i=-\infty}^{\infty} a(k-l) \sum_{i=-\infty}^{\infty} f_i(k) h_{l-i}(k) \\ &= \sum_{i=-\infty}^{\infty} \delta(l) a(k-l) \\ &= a(k)\end{aligned} \tag{1.2.3}$$

该式表明，由式(1.2.2)定义的逆滤波器 $\boldsymbol{f}(k)$可以正确恢复原发数据序列$\{a(k)\}$，从这个意义上讲，它是理想的逆滤波器。然而，理想逆滤波器具有无穷多个抽头，这在实际使用中是不现实的。假定用一个长度为 $2L+1$ 的逆滤波器表示截尾的理想逆滤波器，则该滤波器输出为

$$z(k) = \sum_{i=-L}^{L} f_i(k) y(k-i) \tag{1.2.4}$$

这样就得到可用横向滤波器近似实现逆滤波器，其结构如图 1.5 所示，并且这种近似会产生一部分残余的码间干扰。

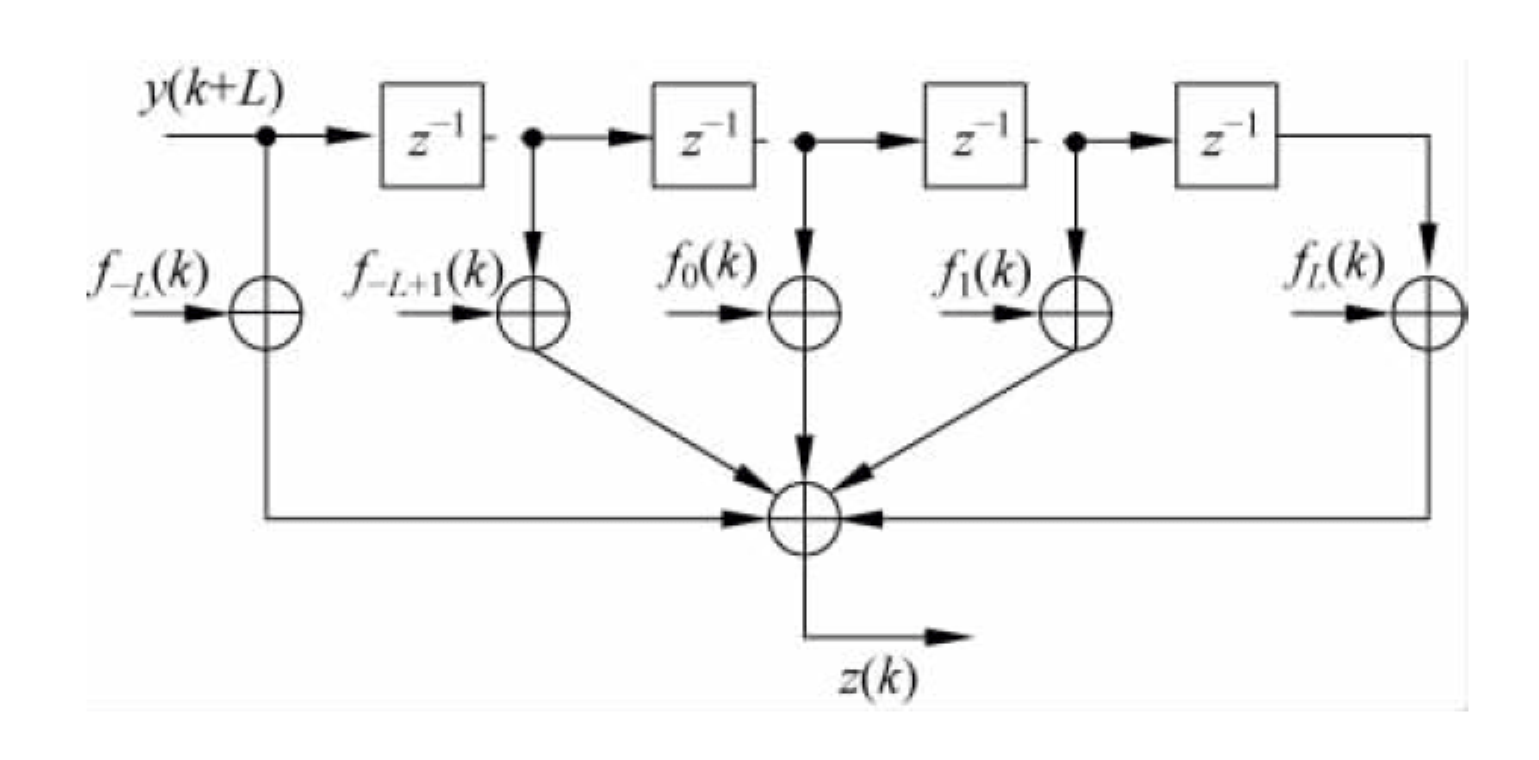



图 1.5 横向滤波器结构

下面分析对无记忆非线性函数 $\Psi\{\cdot\}$的要求。由式(1.2.1)中的滤波器权系数迭代公式可知，当

$$E\{e(k)y(k-i)\} = E\{[\Psi\{z(k)\} - z(k)]y(k-i)\} = 0 \tag{1.2.5}$$

时，横向滤波器权向量 $\boldsymbol{f}(k)$趋于收敛。因此，收敛条件为

$$E\{\Psi(z(k))y(k-i)\} = E\{z(k)y(k-i)\} \quad (\text{对大的 } k, i = -N, \cdots, N) \tag{1.2.6}$$

用 $f_i(k)$同乘式(1.2.6)两边，得

$$E\{\Psi\{z(k)\}\sum_{i=-N}^{N} f_i(k)y(k-i)\} = E\{z(k)\sum_{i=-N}^{N} f_i(k)y(k-i)\} \quad (\text{对大的 } N) \tag{1.2.7}$$

注意到由式(1.2.4)，有

$$z(k-m) = \sum_{i=-N}^{N} f_i(k-m)y(k-m-i) \quad (\text{对大的 } N) \tag{1.2.8}$$

将式(1.2.8)代入式(1.2.6)，则 $\Psi\{\cdot\}$应该满足的条件可以简化为

$$E\{z(k)z(k-m)\} = E\{\Psi\{z(k)\}z(k-m)\} \quad (\text{对大的 } n \text{ 和大的 } N) \tag{1.2.9}$$

**【定义 1.1】** (Bussgang 过程) 若随机过程$\{z(k)\}$满足条件

$$E\{z(k)z(k+m)\} = E\{\Psi\{z(k)\}z(k+m)\} \tag{1.2.10}$$

则称 $z(k)$为 Bussgang 过程。式中，$\Psi\{\cdot\}$为无记忆非线性函数，$m$，$n$ 为整数。

该定义表明，Bussgang过程的自相关函数等于该过程与用它作自变量的无记忆非线性函数之间的互相关。大量的随机过程都属于Bussgang过程，例如高斯过程及所有具有指数衰减自相关函数的随机过程。

**【定义1.2】** （亚高斯和超高斯随机变量） 若随机变量 $X$ 的概率密度函数为

$$p_X(x)=K\exp(-|x/\beta|^{\rho}) \tag{1.2.11}$$

当 $\rho>2$，$\rho=2$ 和 $\rho<2$ 时，分别称 $X$ 为亚高斯、高斯和超高斯随机变量。

**【定义1.3】** 基于Bussgang过程的盲均衡算法称为Bussgang算法。

由于采用 $\Psi\{z(k)\}$ 作为期望信号，因而代价函数为

$$J(k)=E[e^2(k)]=E[|\Psi\{z(k)\}-z(k)^2|] \tag{1.2.12}$$

这时，盲均衡权向量的迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu\hat{\nabla}_f J \tag{1.2.13}$$

式中，$\hat{\nabla}_f J$ 为 $J(k)$ 的梯度估计值，$\mu$ 是一个常数，$\Psi\{z(k)\}$ 取不同的形式，就能得到不同的算法。

关于Bussgang算法的收敛性，由Benveniste-Goursal-Ruget定理来描述。

**Benveniste-Goursal-Ruget定理：** 若数据序列 $\{a(k)\}$ 是亚高斯的，并且 $\psi(z)=\Psi\{z\}-z$ 的二阶导数为负值，即

$$\frac{\partial^2\psi}{\partial z^2}<0,\quad 0<z<\infty \tag{1.2.14}$$

则Bussgang算法是收敛的。

需要指出的是，上面讨论的Bussgang自适应均衡算法只适用于由实基带描述的 $M$ 进制脉冲幅度调制（MPAM）系统。

### 1.2.2 复基带信道的Bussgang算法

现在考虑正交幅度调制QAM系统的自适应均衡，由于QAM系统为幅度调制和相位调制的混合，所以这种系统由复基带信道描述。

在复基带信道中，信源信号、信道冲激响应和接收信号均为复数，分别表示为

$$a(k)=a_{\mathrm{R}}(k)+\mathrm{j}a_{\mathrm{I}}(k) \tag{1.2.15}$$

$$h(k)=h_{\mathrm{R}}(k)+\mathrm{j}h_{\mathrm{I}}(k) \tag{1.2.16}$$

$$y(k)=y_{\mathrm{R}}(k)+\mathrm{j}y_{\mathrm{I}}(k) \tag{1.2.17}$$

式中，下标 R 表示同相分量；I 表示正交分量。

类似地，若给定均衡器输出信号为 $z(k)$，复数据序列 $a(k)$ 的条件均值估计记为 $\hat{a}(k)$，则 $\hat{a}(k)$ 的复基带形式为

$$\hat{a}(k)=E\{a(k)\mid z(k)\}=\hat{a}_{\mathrm{R}}(k)+\mathrm{j}\,\hat{a}_{\mathrm{I}}(k)=\Psi\{z_{\mathrm{R}}(k)\}+\mathrm{j}\Psi\{z_{\mathrm{I}}(k)\} \tag{1.2.18}$$

该式表明，发送数据 $a(k)$ 的同相分量和正交分量，可以由均衡器输出 $z(k)$ 的同相分量和正交分量分别估计。应当注意，只有当同相和正交分量相互统计独立时，条件均值 $E\{a(k)|z(k)\}$ 才能表示成式(1.2.18)的最终形式。复基带信道的 Bussgang 算法的过程如表 1.1 所示。

表 1.1 复基带信道的 Bussgang 算法

| 步　　骤 | 操 作 内 容 |
|---|---|
| 参数选取 | $L$ 为滤波器长度；$\mu$ 为步长因子 $0<\mu<1$；信噪比 |
| 初始化 | 令 $k=0, f_i(0)=\begin{cases}1, & i=0\\ 0, & i=\pm1,\pm2,\cdots,\pm L\end{cases}$ |
| 运算过程 | 对于 $k\geqslant0$，获取数据 $\boldsymbol{y}(k)$<br>$z(k)=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k)=\sum_{i=-L}^{L}f_i(k)y(k-i)$<br>$\hat{a}(k)=\hat{a}_{\mathrm{R}}(k)+\mathrm{j}\,\hat{a}_{\mathrm{I}}(k)=\Psi\{z_{\mathrm{R}}(k)\}+\mathrm{j}\Psi\{z_{\mathrm{I}}(k)\}$<br>$e(k)=\hat{a}(k)-z(k)$<br>$f_i(k+1)=f_i(k)+\mu e(k)y^*(k-i)$ |

### 1.2.3 三种经典的算法

Bussgang 算法有三个非常著名的特例[4-6]：判决引导(DD)算法、Sato

算法和 Godard 算法。

1. 判决引导算法

判决引导(decision directed,DD)算法是 Lucky 于 20 世纪 60 年代提出的,该算法的基本思想是:当误判率足够小时,自适应最小均方(least mean square,LMS)算法中的参考信号可由其估计值代替,即由判决器的输出代替。该算法比较典型的应用是由其他收敛能力较强的盲均衡算法作冷启动,当判决错误率达到足够低的水平时,切换到 DD-LMS 算法,切换门取决于发射信号的形成。图 1.6 表示了工作在判决引导模式的均衡器的方框图。判决引导模式也使用了无记忆非线性函数,它是“量化判决器”。

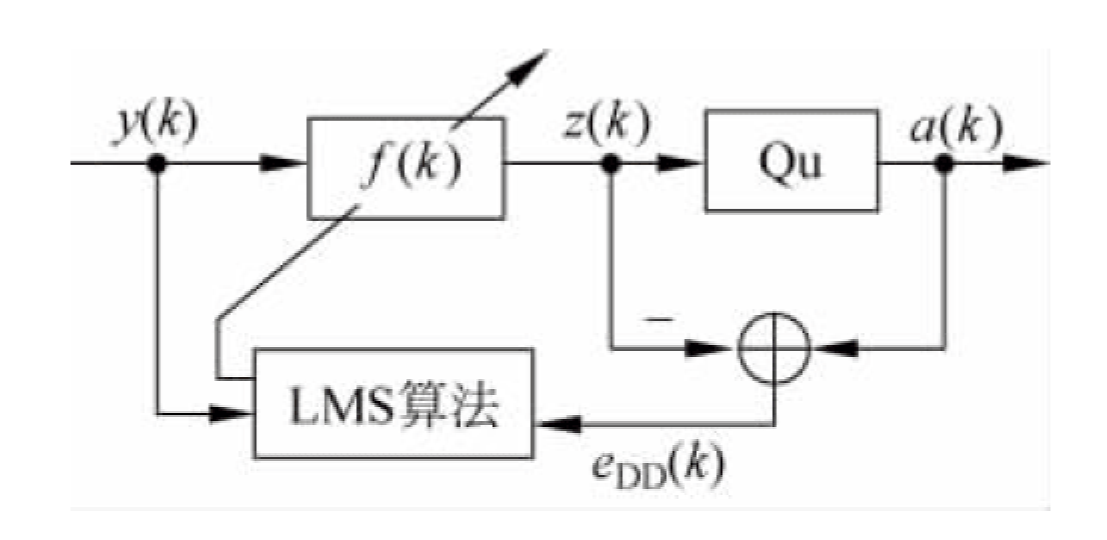



图 1.6 DD 均衡器结构

图 1.6 中,Qu 表示“量化判决”对 $z(k)$ 做出判断,判决值在被发射的数据序列的已知字符中,并使判决结果$\hat{a}(k)$与 $z(k)$最接近,即

$$\hat{a}(k) = \mathrm{dec}[z(k)] \tag{1.2.19}$$

式中,dec[·]表示判决操作。

例如,在二进制等概率数据序列的简单情况下,数据和判决值分别为

$$a(k) = \begin{cases} +1, & \text{对字符 } 1 \\ -1, & \text{对字符 } 0 \end{cases} \tag{1.2.20}$$

和

$$\mathrm{dec}\{z(k)\} = \mathrm{sgn}\{z(k)\} \tag{1.2.21}$$

DD 算法的代价函数为

$$J_{\mathrm{DD}} = E\left\{\frac{1}{2}[\hat{a}(k) - z(k)]^2\right\} \tag{1.2.22}$$

误差信号为

$$e_{\mathrm{DD}}(k) = \hat{a}(k) - z(k) \tag{1.2.23}$$

DD 算法权向量迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu[\hat{a}(k) - z(k)]\boldsymbol{y}^*(k) \tag{1.2.24}$$

将 DD 算法与 Bussgang 算法比较可知，DD 算法是取 $g\{\cdot\}=\mathrm{sgn}\{\cdot\}$的 Bussgang 算法。

2. Sato 算法

Sato 提出了一种盲均衡算法，用来均衡多种电平脉冲幅度调制（pulse amplitude modulation，PAM）信号，其性能比 DD 算法更稳健。该算法代价函数为

$$J(k) = E\{(|z(k)| - R)^2\} \tag{1.2.25}$$

式中，$R$ 为发射信号统计模值，定义为

$$R = \frac{E\{|a(k)|^2\}}{E\{a(k)\}} \tag{1.2.26}$$

该算法中的无记忆非线性估计值为

$$\hat{a}(k) = R\mathrm{csgn}[z(k)] \tag{1.2.27}$$

$$e_{\mathrm{s}}(k) = R\mathrm{csgn}[z(k)] - z(k) \tag{1.2.28}$$

式中

$$\mathrm{csgn}(x) = \mathrm{csgn}(x_{\mathrm{R}} + \mathrm{j}x_{\mathrm{I}}) = \mathrm{sgn}(x_{\mathrm{R}}) + \mathrm{j}\mathrm{sgn}(x_{\mathrm{I}}) \tag{1.2.29}$$

Sato 算法权向量迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu\{R\mathrm{csgn}[z(k)] - z(k)\}\boldsymbol{y}^*(k) \tag{1.2.30}$$

Sato 算法是 Bussgang 算法取 $g\{z(k)\}=R\mathrm{sgn}[z(k)]$时的一个特例。且当 $R=1$ 时 Sato 算法退化为 DD 算法。

3. Godard 算法

Godard 最早提出了常数盲均衡算法，适用于对所有恒定包络（常数模）的发射信号进行均衡。该算法的代价函数为

$$J(k)=E\{(|z(k)|^p-R^p)^2\},\quad p=1,2 \tag{1.2.31}$$

式中，$R^p$ 为发射信号统计模型，定义为

$$R^p=\frac{E\{|a(k)|^{2p}\}}{E\{|a(k)|^p\}} \tag{1.2.32}$$

Godard 算法权向量迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu z(k)|z(k)|^{p-2}(|z(k)|^p-R^P)\boldsymbol{y}^*(k) \tag{1.2.33}$$

当 $p=1$ 时，Godard 算法表示为 CMA1-2，且与 Sato 算法相同；当 $p=2$，Godard 算法就是常数模算法，表示为 CMA2-2，简记为 CMA，它适用于对恒定模值的发射信号均衡。

DD 算法、Sato 算法和 Godard 算法都是对均衡器输出进行非线性变换来产生期望信号的。无记忆非线性变换函数的表达式总结如表 1.2 所示。

表 1.2　三种 Bussgang 算法的无记忆非线性函数

| 算　　法 | 无记忆非线性函数 |
|---|---|
| DD | $\mathrm{sgn}(\cdot)$ |
| Sato | $R\mathrm{csgn}(\cdot)$ |
| Godard | $z(k)(\|z(k)\|+R^p\|z(k)\|^{p-1}-\|z(k)\|^{2p-1})/\|z(k)\|$ |

## 1.2.4　常数模算法及其性能

### 1. 常数模算法

CMA 是 Godard 算法的特例（$p=2$），利用（隐含的）高阶统计特性构造代价函数，通过调整均衡器的权向量寻找代价函数的极值点，其构造如图 1.7 所示。

图 1.7 中，$\boldsymbol{a}(k)$是零均值独立同分布的发射信号向量；$\boldsymbol{h}(k)$是信道脉冲响应向量且长度为 $M$，即 $\boldsymbol{h}(k)=[h_0(k),\cdots,h_{M-1}(k)]^{\mathrm{T}}$（上标 T 表示转

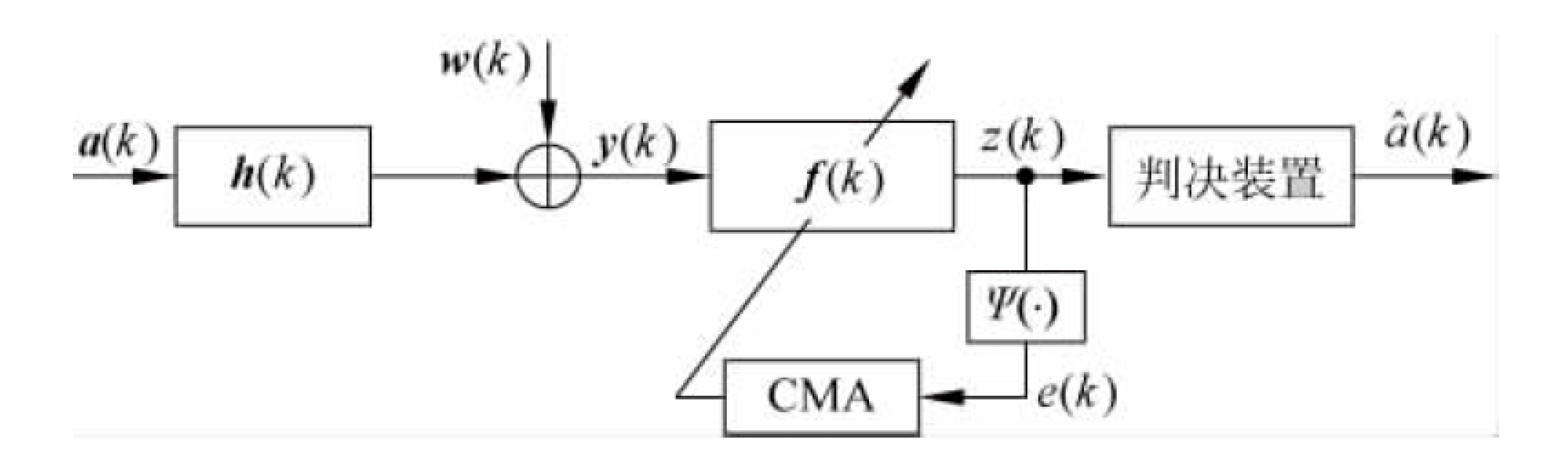



图 1.7 CMA 结构

置)；$\boldsymbol{w}(k)$是加性高斯白噪声；$\Psi(\cdot)$是误差生成函数；$e(k)$是误差函数；$\boldsymbol{f}(k)=[f_{-L}(k),\cdots,f_0(k),\cdots,f_L(k)]^{\mathrm{T}}$ 为均衡器权向量且长度为 $2N+1$；$\boldsymbol{y}(k)$是均衡器接收信号向量；$z(k)$是均衡器输出信号；$\hat{a}(k)$是判决装置对 $z(k)$的判决输出信号。

设 $\boldsymbol{a}(k)=[a(k),\cdots,a(k-M+1)]^{\mathrm{T}}$，$\boldsymbol{y}(k)=[y(k+L),\cdots,y(k),\cdots,y(k+L)]^{\mathrm{T}}$，则由图 1.7，得

$$y(k)=\sum_{i=0}^{M-1}h_i(k)a(k-i)+w(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+w(k) \tag{1.2.34}$$

$$z(k)=\sum_{i=-L}^{L}f_i(k)y(k-i)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)=\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{1.2.35}$$

CMA 误差函数为

$$e(k)=|z(k)|^2-R^2 \tag{1.2.36}$$

式中，$R^2$ 为 CMA 模值，定义为

$$R^2=\frac{E\{|a(k)|^4\}}{E\{|a(k)|^2\}} \tag{1.2.37}$$

CMA 代价函数为

$$J=\frac{1}{2}E[e^2(k)] \tag{1.2.38}$$

为了最小化 CMA 代价函数，采用随机梯度法对 $\boldsymbol{f}(k)$进行调整，即

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu\hat{\nabla}_f J \tag{1.2.39}$$

式中，$\mu$ 表示步长，且为较小的正数；$\hat{\nabla}_f J$ 为 $J$ 对 $\boldsymbol{f}(k)$求偏导后取瞬时值。

由 $J$ 对 $\boldsymbol{f}(k)$求偏导得

$$\hat{\nabla}_f = \frac{\partial J}{\partial \boldsymbol{f}(k)}$$
$$= E[2(|z(k)|^2 - R^2)] \frac{\partial[z(k)z^*(k)]}{\partial \boldsymbol{f}(k)}$$
$$= E[2(|z(k)|^2 - R^2)] \frac{\partial[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)(\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k))^*]}{\partial \boldsymbol{f}(k)}$$
$$= E[4(|z(k)|^2 - R^2)\boldsymbol{y}^*(k)\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k)]$$
$$= E[4(|z(k)|^2 - R^2)\boldsymbol{y}^*(k)z(k)] \qquad (1.2.40)$$

$$\hat{\nabla}_f J = 2(|z(k)|^2 - R^2)\boldsymbol{y}^*(k)z(k) \qquad (1.2.41)$$

将式(1.2.36)和式(1.2.41)代入式(1.2.39)，得 CMA 权向量迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - 2\mu(|z(k)|^2 - R^2)\boldsymbol{y}^*(k)z(k)$$
$$= \boldsymbol{f}(k) - 2\mu e(k)z(k)\boldsymbol{y}^*(k) \qquad (1.2.42)$$

2. 常数算法性能

(1) CMA 均衡器与 Wiener 均衡器的相似性能

在需要借助训练序列的自适应均衡算法中，若代价函数的解唯一，则称为 Wiener 解。期望不需要借助训练序列的 CMA，同样具有与 Wiener 解相似的性能。如果发射信号为 BPSK（二进制相移键控）实信号，即 $a(k)=\pm 1$，训练序列的延迟为 $d$，则可以证明 Wiener 均衡器的代价函数 $J_{\mathrm{W}}$ 和 CMA 代价函数 $J_{\mathrm{CMA}}$之间的近似[7]关系为

$$J_{\mathrm{CMA}} \approx J_{\mathrm{W}}^{\min}\left[1 + \frac{9}{4}J_{\mathrm{W}}^{\min}\right] \qquad (1.2.43)$$

式中，$0 \leqslant J_{\mathrm{W}}^{\min} \leqslant 0.25$。常数模均衡器权向量 $\boldsymbol{f}_{\mathrm{CMA}}(k)$和 Wiener 均衡器权向量 $\boldsymbol{f}_{\mathrm{opt}}(k)$的比值为

$$\alpha = \frac{f_{\mathrm{CMA}}(k)}{f_{\mathrm{opt}}(k)} \approx 1 - \frac{2}{3}J_{\mathrm{W}}^{\min} \qquad (1.2.44)$$

结果表明，Wiener 均衡器和 CMA 均衡器权向量的差值取决于 CMA 误差函数的大小，并且当 CMA 均衡器具有良好的性能时，$e(k)$很小且

$\boldsymbol{f}_{\mathrm{CMA}}(k)$接近 $\boldsymbol{f}_{\mathrm{opt}}(k)$。

(2) 常数模算法代价函数的局部极小值与初始化[8]

常数模盲均衡代价函数相对于权向量而言,存在局部极小值。常数模算法采用不同的初始化形式,会收敛到不同的局部极小值,这与自适应LMS 算法有所不同。下面给出一个仿真例子,观察二维权向量情况下,CMA 与 LMS 算法采用不同权向量初始化时的收敛情况以及代价函数的极小值分布。

简单起见,采用一单极点信道

$$H(z)=\frac{1}{1-0.6z^{-1}} \tag{1.2.45}$$

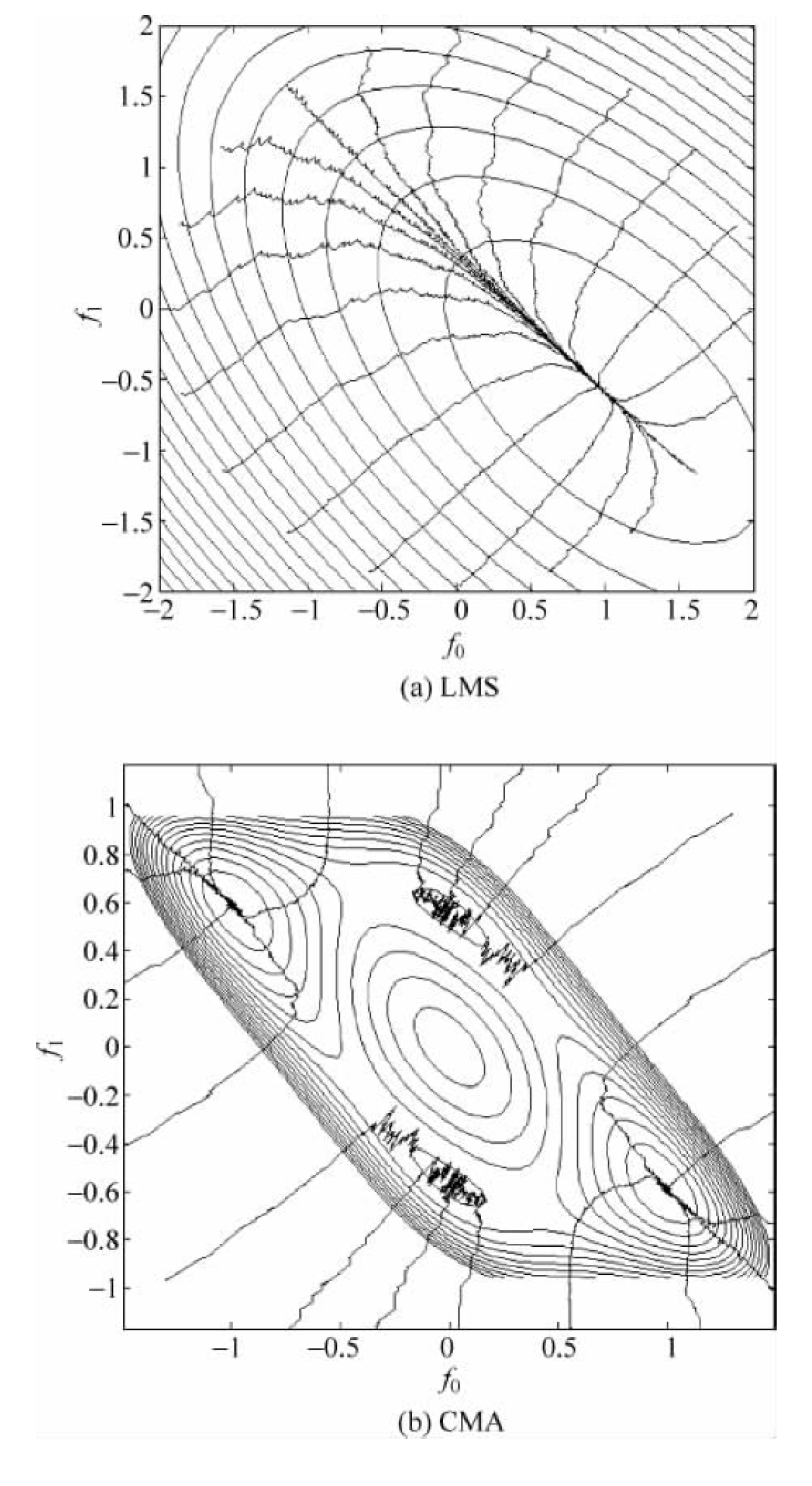
对于这样的信道,波特间隔均衡器容易获得较好的均衡效果。选择20 组二维权向量 $\boldsymbol{f}=[f_0,f_1]$,从代价函数性能表面的不同方向来初始化均衡器。仿真结果如图 1.8 所示。

从图 1.8(a)可以看出,若采用 LMS 算法自适应求解均衡器参数,则不管初始化形式如何,均衡器权头系数都收敛到了它的真值解 $\boldsymbol{f}_{\mathrm{true}}=[1,-0.6]^{\mathrm{T}}$。这表明 LMS 算法只有一个最小值点,即全局最优点。而图 1.8(b)表明,对于 CMA,权向量收敛到了 4 个不同的极小值点,不同位置的权向量初始化会收敛到不同的极小值。其中,一个极小值为真正的信道的逆,即 $\boldsymbol{f}_{\mathrm{true}}=[1,-0.6]^{\mathrm{T}}$;另一个与真正解符号相反,如果采用差分编码,该解也是可以接受的。其他的两个极小值都不是信道的逆。因此,CMA 的标准代价函数是多模态的,即存在多个极小值。

大量研究及工程经验表明,当权向量足够长时,采用中心抽头初始化(权向量中间一个抽头为 1,其余抽头都为 0),CMA 总能够收敛到全局最优解。但是均衡器抽头的初始化问题至今并未从理论上完全解决,很难确定使用哪一种方式更好。因此,CMA(所有 Bussgang 类盲均衡算法)的初始化问题仍然是个待研究的问题。

### 1.2.5 分数间隔盲均衡算法

按采样率划分,均衡器分为波特间隔均衡器和分数间隔均衡器。波特



(a) LMS

(b) CMA

图 1.8 收敛特性

间隔均衡器使用码率或波特率对输入输出进行采样；而分数间隔均衡器(fractionally spaced equalizer，FSE)是对输入输出以比码率或波特率更高的速率进行采样的，能够降低对时间相位的敏感性和噪声放大，具有一些重要的特点：①对于一些单位圆上有零点的信道或具有深度谱零点的信道进行均衡，不需要大量参数，可以快速收敛；②在满足某些条件下，使用有限脉冲响应(finite impulse response，FIR)均衡器就可实现完全均衡(perfect equalization，PE)。"完全均衡"是指对于某个固定的非负整数延迟 $\tau$ 和任意源序列 $\{a(k)\}$，均衡器的输出 $z(k)$ 满足条件：$z(k)=a(k-\tau)$。这一响应表示系统只将发射的码简单地延迟了 $\tau$ 个波特距离。

本节首先讨论采样率理论，再分析分数间隔盲均衡器的一般结构。

1. 多采样率理论[9,10]

抽头间隔为码元间隔 $T$ 的均衡器，使用码率或波特率 $1/T$ 对输入输出信号进行采样。而分数间隔均衡器的输入是通过对信道输出信号用比码率更高的速率进行采样而得来的，其主要思想就是对接收信号过采样，以获得更多、更详细的信道信息。分数间隔均衡器最具吸引力的特性是：在理想情况下，有限脉冲响应的 FSE 能完全均衡一个 FIR 信道。

(1) 过采样

多采样包括欠采样和过采样两种。欠采样是从信号 $a(i)$ 的每 $P$ 个采样值中取出一个，而使采样率降低到 $1/P$。分数间隔均衡器是增加采样率，属过采样。

在信号 $a(i)$ 的每两个采样值之间插入 $P-1$ 个零值，可以使其采样率增加 $P$ 倍。通常，还需要后接一个低通滤波器 $h(k)$，如图 1.9 所示。

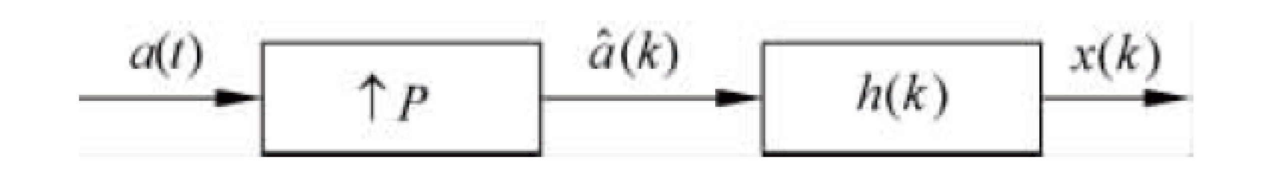



图 1.9 过采样率框图

过采样的输入-输出关系为

$$\tilde{a}(k)=\sum_{i=-\infty}^{+\infty}a(i)\delta(k-iP)=\begin{cases}\tilde{a}(k/P), & k=0,\pm1,\cdots,\pm2L,\cdots\\0, & \text{其他}\end{cases}\tag{1.2.46}$$

整个系统的输入输出关系为

$$x(k)=\sum_{i=-\infty}^{+\infty}h_i(k)a(k-iP)\tag{1.2.47}$$

(2) 多速率系统模型

具有 $T/P$ 间隔发送机-接收机的多速率系统模型,如图 1.10 所示。

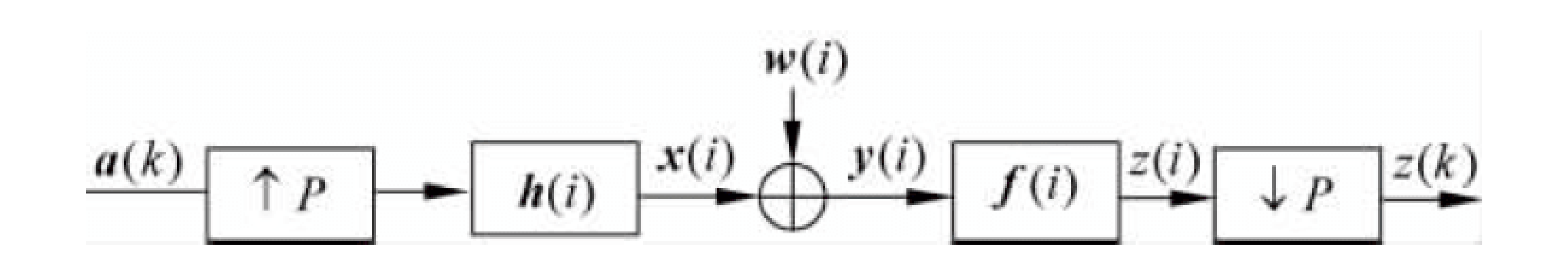



图 1.10 $T/P$ 间隔均衡器的多速率系统模型

图中,$\uparrow P$ 和 $\downarrow P$ 分别表示内插和抽取操作,$\{a(k)\}$是以符号长度 $T$ 为周期的发射信序列,$\boldsymbol{h}(k)$是 $T/P$ 间隔信道冲激响应向量(发射滤波器、未知信道和接收滤波器三者合成的信道),长度为 $P\times N_c$($N_c$ 表示符号间隔抽样的信道冲激响应长度);$\boldsymbol{f}(i)$表示 $T/P$ 间隔均衡器权向量,长度为 $P\times N_f$($N_f$ 表示符号间隔抽样的均衡器响应长度),$\boldsymbol{w}(k)$是信道噪声的 $T/P$ 间隔抽样值,信道输出信号经过 $P$ 倍的速率进行抽取,便得到分数间隔信道输出序列$\{y(i)\}$。

$$y(i)\triangleq y\left(i\frac{T}{P}\right)=\sum_{l=-\infty}^{+\infty}a(l)h\left(i\frac{T}{P}-lT\right)+w\left(i\frac{T}{P}\right)\tag{1.2.48}$$

该信号经过抽头系数序列为$\{f(i)\}$的均衡器后,输出 $z(k)$表示为

$$z(k)=\sum_i f_i(k)y(k-i)\tag{1.2.49}$$

## 2. 多信道系统模型及其性能分析

(1) 多信道系统模型

与多速率系统模型等效的另一种十分有用的结构模型是多信道系统

模型。$T/P$ 分数间隔多信道系统模型如图 1.11 所示。

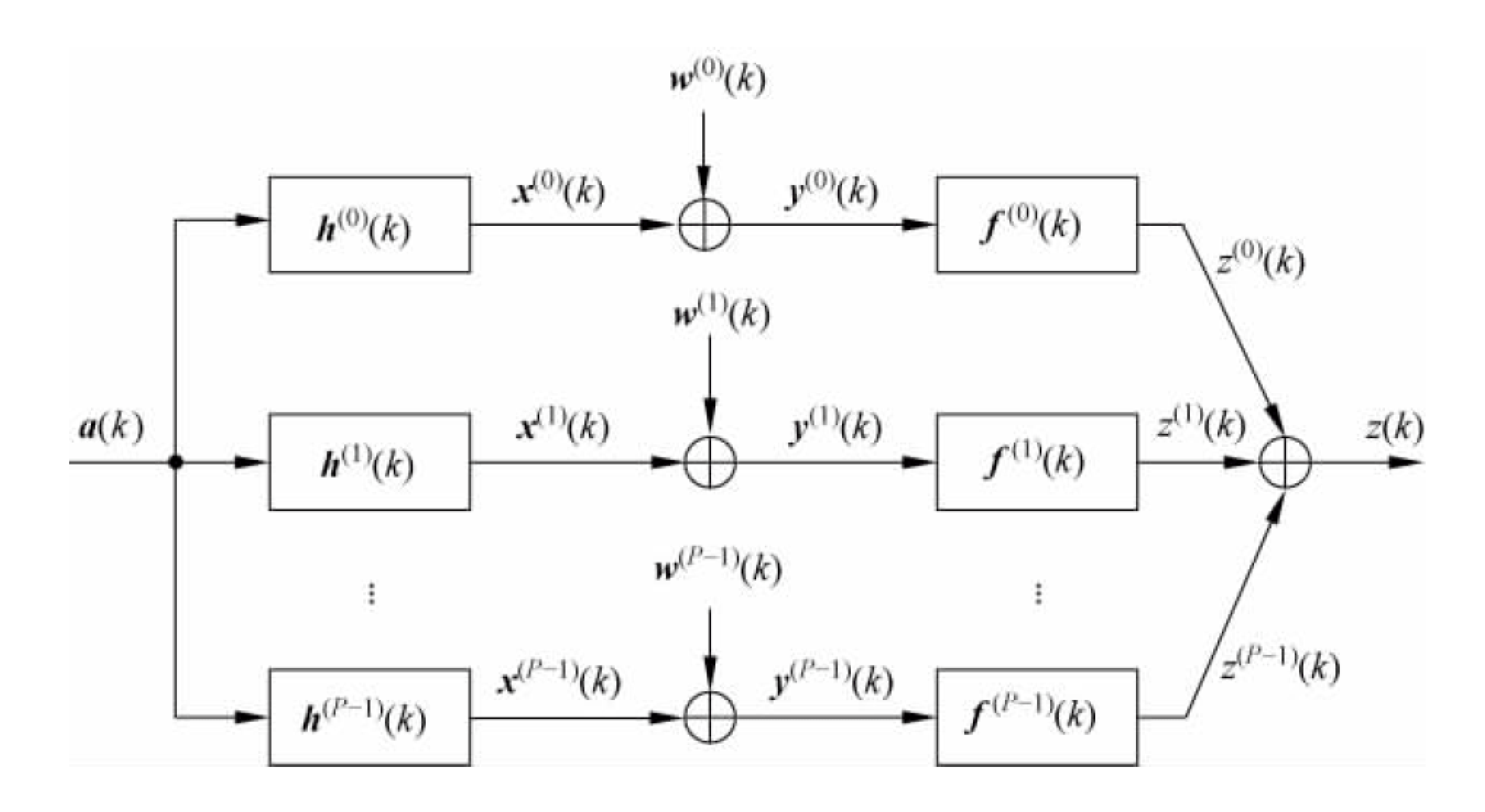



图 1.11 $T/P$ 分数间隔多信道均衡模型

多信道系统模型可以用单一的符号速率表示信号的传输过程,消除内插和抽取这样的变速处理,简化了分析过程。这种系统模型在结构上也比较简单,便于后续的分析研究。多信道系统模型中子信道、子均衡器的权向量、各个输入输出序列与多速率系统模型的分数间隔信道、均衡器、各个输入输出序列之间存在着如下的对应关系。

第 $p$ 个子信道($p=0,1,\cdots,P-1$)的冲激响应为

$$h^{(p)}(k)=h[(k+1)P-p-1] \tag{1.2.50}$$

子信道的输出为

$$y^{(p)}(k)=\sum_{i=0}^{M-1}h_i^{(p)}(k)a(k-i)+w^{(p)}(k) \tag{1.2.51}$$

定义三个波特率表示的向量

$$\boldsymbol{h}(k)=\begin{bmatrix}\boldsymbol{h}^{(0)}(k)\\ \vdots\\ \boldsymbol{h}^{(P-1)}(k)\end{bmatrix},\quad \boldsymbol{y}(k)=\begin{bmatrix}y^{(0)}(k)\\ \vdots\\ y^{(P-1)}(k)\end{bmatrix},\quad \boldsymbol{w}(k)=\begin{bmatrix}w^{(0)}(k)\\ \vdots\\ w^{(P-1)}(k)\end{bmatrix} \tag{1.2.52}$$

它们之间的关系为

$$y(k) = \sum_{i=0}^{M-1} h_i(k)a(k-i) + w(k) \tag{1.2.53}$$

将式(1.2.53)的接收信号进一步扩展为

$$\underbrace{\begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-L+1) \end{bmatrix}}_{\boldsymbol{y}(k)} = \underbrace{\begin{bmatrix} h(0) & h(1) & \cdots & h(M-1) & & & & 0 \\ & h(0) & h(1) & \cdots & h(M-1) & & & \\ 0 & & \ddots & \ddots & \ddots & & & \\ & & & h(0) & & h(1) & \cdots & h(M-1) \end{bmatrix}}_{\boldsymbol{h}} \cdot \underbrace{\begin{bmatrix} a(k) \\ a(k-1) \\ \vdots \\ a(k-M-L+1) \end{bmatrix}}_{\boldsymbol{a}(k)} + \underbrace{\begin{bmatrix} w(k) \\ w(k-1) \\ \vdots \\ w(k-L+1) \end{bmatrix}}_{\boldsymbol{w}(k)} \tag{1.2.54}$$

即

$$\boldsymbol{y}(k) = \boldsymbol{ha}(k) + \boldsymbol{w}(k) \tag{1.2.55}$$

式中,$\boldsymbol{h}$ 称作信道矩阵。

设第 $p$ 个子信道 $h^{(p)}(k)$对应的均衡器与多抽样率的关系是 $f^{(p)}(k) = f(kP+p)$,则分数间隔均衡器的输出为

$$z(k) = \sum_{i=0}^{L-1} f_i^{\mathrm{T}}(k)y(k-i) = \boldsymbol{f}^{\mathrm{T}}\boldsymbol{y}(k) \tag{1.2.56}$$

式中

$$\boldsymbol{f}(k) = \begin{bmatrix} f^{(0)}(k) \\ \vdots \\ f^{(P-1)}(k) \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} f_0 \\ \vdots \\ f_{L-1} \end{bmatrix}, \quad \boldsymbol{y}(k) = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-L+1) \end{bmatrix} \tag{1.2.57}$$

将式(1.2.54)代入式(1.2.55),得

$$y(k) = \boldsymbol{f}^{\mathrm{T}}[\boldsymbol{ha}(k) + \boldsymbol{w}(k)] = \boldsymbol{f}^{\mathrm{T}}\boldsymbol{ha}(k) + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{w}(k) = \boldsymbol{c}^{\mathrm{T}}(k)\boldsymbol{a}(k) + v(k) \tag{1.2.58}$$

式中，$\boldsymbol{c}=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{f}$ 表示系统(信道-均衡)冲激响应，它的系数是以波特率采样的样点；$\boldsymbol{v}(k)=\boldsymbol{f}^{\mathrm{T}}\boldsymbol{w}(k)$是系统输出噪声。

(2) 多信道系统模型性能分析

波特间隔均衡器对输入和输出信号都以 $1/T$ 的速率采样，所以从频域角度看，波特间隔均衡器输入信号的频谱可以写为

$$
\begin{aligned}
\boldsymbol{Y}_T(\omega) &= \frac{1}{T}\sum_{i=-\infty}^{\infty} a(k)AH\left(\omega-\frac{2\pi i}{T}\right)\exp\left[\mathrm{j}\left(\omega-\frac{2\pi i}{T}\right)t_0\right] \\
&= \frac{1}{T}\sum_{i=-\infty}^{\infty} X\left(\omega-\frac{2\pi i}{T}\right)\exp\left[\mathrm{j}\left(\omega-\frac{2\pi i}{T}\right)t_0\right]
\end{aligned}
\tag{1.2.59}
$$

式中，$\mathrm{j}=\sqrt{-1}$为虚数单位，$X(\omega)=a(k)AH(\omega)$。波特间隔均衡器结构中，采样速率 $1/T$ 小于 Nyquist 采样速率 $2/T$，所以 $Y_T(\omega)$为折叠或混叠频谱，折叠频率为 $1/2T$。波特均衡器输出信号的频谱为 $F_T(\omega)Y_T(\omega)$，加上、下标表示采样间隔为码元间隔 $T$，$F_T(\omega)$是均衡器的频率特性，且

$$
F_T(\omega) = \sum_{k=0}^{L-1} f(k)\exp[-\mathrm{j}\omega kT] \tag{1.2.60}
$$

由此可见，在频域中，由于接收信号频谱的混叠，波特间隔均衡器只能补偿接收信号 $y(k)$混叠的频率响应特性，而信道的频率特性 $H(\omega)$由于混叠而损失掉了，所以，均衡器不可能补偿信道的频率畸变。

分数间隔均衡器对输入输出信号都以 $T'=T/P$($P>1$ 且为整数)的速率采样，从频率域角度看，分数间隔均衡器的频率特性为

$$
F_T(\omega) = \sum_{k=0}^{L-1} f(k)\exp[-\mathrm{j}\omega kT'] \tag{1.2.61}
$$

信号经过均衡器的频谱可表示为

$$
F_{T'}(\omega)Y_{T'}(\omega) = \frac{1}{T'}F_{T'}(\omega)\sum_{i=-\infty}^{\infty} X\left(\omega-\frac{2\pi i}{T'}\right)\exp\left[\mathrm{j}\left(\omega-\frac{2\pi i}{T'}\right)t_0\right] \tag{1.2.62}
$$

当$|\omega|>\dfrac{2\pi P}{T}$时，$X(\omega)=0$，所以式(1.2.61)可进一步表示为

$$
F_{T'}(\omega)Y_{T'}(\omega) = \frac{1}{T'}F_{T'}(\omega)X(\omega)\exp[\mathrm{j}\omega t_0], \quad |\omega|\leqslant\frac{\pi}{T'} \tag{1.2.63}
$$

由此可见，由于对分数间隔均衡器输入信号的采样频率提高了，因而避免了因欠采样引起的频谱混叠，接收信号频谱中含有信道的频率特性函数，均衡器可以有效地补偿信道特性的畸变。分数间隔均衡器的输出信号送到判决器进行判决，码率仍然是 $1/T$，即

$$\frac{1}{T'}\sum_{i=-\infty}^{\infty}F_{T'}\left(\omega-\frac{2\pi i}{T}\right)X\left(\omega-\frac{2\pi i}{T}\right)\exp\left[\mathrm{j}\left(\omega-\frac{2\pi i}{T}\right)t_0\right]$$

实际上，分数间隔均衡器等效于一个匹配滤波器加上一个码元间隔均衡器。

### 1.2.6 超指数盲均衡算法

#### 1. 超指数算法

Shalvi 和 Weinstein[11] 提出了一种基于高阶统计量的盲均衡算法，由于其收敛过程中对码间干扰的抑制近似于超指数规律（相对于功率而言），所以称为超指数（super exponential，SE）算法。该算法通过求取均衡器序列自相关矩阵和均衡器输入输出序列之间的四阶互累积量求解权向量。在该算法中，令

$$\boldsymbol{f}'=\boldsymbol{R}^{-1}\boldsymbol{d} \tag{1.2.64}$$

式中，上标$[\cdot]^{-1}$表示矩阵求逆运算。权向量

$$\boldsymbol{f}=\frac{1}{\sqrt{(\boldsymbol{f}')^{\mathrm{T}}\boldsymbol{R}\boldsymbol{f}'}}\boldsymbol{f}' \tag{1.2.65}$$

式中，$\boldsymbol{R}$ 是 $L\times L$ 维矩阵，其元素为

$$R_{nm}=\frac{\mathrm{cum}[y(k-n);y^*(k-m)]}{\mathrm{cum}[a(k);a^*(k)]} \tag{1.2.66}$$

$\boldsymbol{d}$ 为 $L\times 1$ 维向量，其元素为

$$d_n=\frac{\mathrm{cum}[z(k-\delta_f):p;z^*(k-\delta_f):q;y^*(k-n)]}{\mathrm{cum}[a(k):p;a^*(k):q+1]} \tag{1.2.67}$$

式中，$\mathrm{cum}(\underbrace{z_1;\ z_1;\ \cdots;\ z_1}_{p};\cdots)=\mathrm{cum}(z_1:p;\ \cdots)$。

均衡器输出为

$$z(k) = \boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{1.1.68}$$

式(1.2.67)中,$p$,$q$ 表示随机过程的个数,常用的算法为 $p=2$,$q=1$。与文献[10]不同的是,本书在式(1.2.67)中加入了参数 $\delta_f$,$\delta_f$ 为均衡器输出初始化时延。文献[8]表明,在有限权长的条件下,$\delta_f$ 的取值会影响算法的收敛性能。SE 算法是对均衡器输入输出的数据块进行运算,反复迭代式(1.2.64)～式(1.2.68),其收敛速度很快,一般经过 3～5 次迭代即可收敛。而每次求解都对数据块进行累积量运算,因此计算量较大。对该算法的计算量可作如下估计:假设发射信号为实信号,数据长度为 $M$,权向量长度为 $L$。计算 $\boldsymbol{R}$ 矩阵需 $ML$ 次乘法(一段数据仅计算 1 次),计算式(1.2.68)所示的均衡器输出序列需 $ML$ 次乘法,计算式(1.2.67)的 $d_n$ 需 $(2L+1)M$ 次乘法。另外,计算式(1.2.64)中 $\boldsymbol{R}^{-1}$,$\boldsymbol{R}^{-1}$ 与 $\boldsymbol{d}$ 的乘积,以及式(1.2.65)所示的归一化权向量,均需要 $L^2$ 量级的乘法。相比而言,如果权向量长度 $L$ 远远小于数据长度 $M$,则这些乘法可以忽略。设 $p=2$,$q=1$,并假设算法迭代 5 次收敛,则 SE 算法的计算量约为 $16ML+5M$;而相同数据长度下,CMA 的计算量为 $2ML+3L$。相比之下,SE 算法的计算量要大得多。

2. 超指数迭代算法

文献[10]从超指数算法推导出了超指数迭代(super exponential iteration,SEI)算法。当式(1.2.67)中 $p=2$,$q=1$ 时,所得算法的收敛性能较好。将算法总结在表 1.3 中。表 1.3 中的 $R^2$ 即为 CMA 的模。与 CMA 相比,SEI 算法比 CMA 多了一项计算矩阵 $\boldsymbol{Q}$ 的表达式,而正是该矩阵能对数据起到白化作用,从而使该算法收敛速度比 CMA 快。文献[8]指出,对 SEI 算法作适当简化,即可获得 CMA。从而表明,CMA 隐含地使用了高阶统计量。在 SEI 算法中,计算 $\boldsymbol{Q}$ 矩阵的计算量与权向量长度的平方成正比。

表 1.3 SEI 算法的迭代过程

中心抽头初始化权向量 $\boldsymbol{f}(0)=[0,\cdots,1,\cdots,0]^{\mathrm{T}}$
采用一段数据，计算相关矩阵 $\boldsymbol{R}$，

$$\begin{aligned}
&\boldsymbol{Q}(0)=\boldsymbol{R}^{-1}\\
&z(k)=\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k)\\
&\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu\cdot\boldsymbol{Q}(k)\boldsymbol{y}^{*}(k)z(k)(|z(k)|^{2}-R^{2})\\
&\boldsymbol{Q}(k+1)=\frac{1}{1-\mu}\left[\boldsymbol{Q}(k)-\frac{\mu\boldsymbol{Q}(k)\boldsymbol{y}^{*}(k)\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{Q}(k)}{1-\mu+\mu\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{Q}(k)\boldsymbol{y}^{*}(k)}\right]
\end{aligned}\tag{1.2.69}$$

3. 稀疏权超指数盲均衡算法

通过以上分析可以看出，超指数(SE)算法具有良好的权向量估计性能，且具有很快的收敛速度，但当权向量较长时其计算量较大。对于有稀疏性的信道(所谓稀疏性，指信道能量集中在相隔较远的几个抽头上，而大多数抽头的能量很小)，结合超指数盲均衡算法的收敛特点，利用信道的稀疏特性，可以减小该算法的计算量。

超指数算法需重复几次迭代才能收敛。每次迭代均对互四阶累积量 $\boldsymbol{d}$[式(1.2.67)]和均衡器输出[式(1.2.68)]进行计算。对于稀疏信道，权向量具有稀疏性，此时式(1.2.68)所示的均衡器输出可由稀疏权方法简化其计算量，即保留显著抽头，不显著抽头置零。现分析信道的稀疏性，会不会带来 $\boldsymbol{d}$ 向量也呈现稀疏性。

对于 SE 算法，$p=2,q=1$ 时，由式(1.2.67)，并考虑将均衡器输入延迟为 $\delta_f(0\leqslant\delta_f\leqslant N_f-1)$，初始化均衡器输出，则有

$$\begin{aligned}
&\operatorname{cum}\{y(k-\delta_f);\ y(k-\delta_f);\ y^{*}(k-\delta_f);\ y^{*}(k-i)\}\\
&=\operatorname{cum}\Big\{\sum_{l_1}h_{l_1}(k)a(k-\delta_f-l_1);\ \sum_{l_2}h_{l_2}(k)a(k-\delta_f-l_2);\\
&\quad\sum_{l_3}h_{l_3}^{*}(k)a^{*}(k-\delta_f-l_3);\ \sum_{l_4}h_{l_4}^{*}(k)a^{*}(k-i-l_4)\Big\}\\
&=\sum_{l_1}\sum_{l_2}\sum_{l_3}\sum_{l_4}h_{l_1}(k)h_{l_2}(k)h_{l_3}^{*}(k)h_{l_4}^{*}(k)\operatorname{cum}\{a(k-\delta_f-l_1);\\
&\quad a(k-\delta_f-l_2);\ a^{*}(k-\delta_f-l_3);\ a^{*}(k-i-l_4)\}\\
&=\sum_{l_4}h_{i-\delta_f+l_4}^{2}(k)h_{i-\delta_f+l_4}^{*}(k)h_{l_4}^{*}(k)\operatorname{cum}\{a(k);\ a(k);\ a^{*}(k);\ a^{*}(k)\}
\end{aligned}$$

所以

$$d_i = \sum_{l_k} h^2_{i-\delta_f+l_k}(k) h^*_{i-\delta_f+l_k}(k) h^*_{l_k}(k) \tag{1.2.70}$$

当 $i=\delta_f$ 时，式(1.2.70)为归一化峰度值，而且 $i$ 与 $\delta_f$ 构成了一种确定的时延关系。当 $\delta_f$ 改变时，归一化峰值位置会相应改变，所有元素 $d_i$ 也会作相应平移。文献[8]表明，$\boldsymbol{d}$ 向量延迟与均衡器延迟是一一对应的。因此，改变 $\delta_f$ 可以改变均衡器延迟。由式(1.2.70)，对于稀疏信道，当 $i-\delta_f$ 是较显著的稀疏抽头间隔时，等式右边乘积求和才得较大值，$d_i$ 才较显著。正是由信道稀疏性，使得 $\boldsymbol{d}$ 向量也将呈现稀疏性，这就有可能对 $\boldsymbol{d}$ 向量的计算进行简化。具体做法是，在第一次迭代中对整个 $\boldsymbol{d}$ 向量进行计算之后，根据一个门限，选取高于该门限的元素，其余元素置零。之后，根据式(1.2.64)和式(1.2.65)计算权向量。所得权向量也根据一个门限，保留高于该门限的抽头。在下一次迭代中，仅计算对应较显著的 $\boldsymbol{d}$ 向量中的元素，并再次与门限进行比较筛选，则 $\boldsymbol{d}$ 的计算得到简化。权向量抽头的选取仍采用第一次迭代中的方法。在以后的迭代中重复该过程，直至算法收敛。对于稀疏信道，较显著的 $\boldsymbol{d}$ 向量元素和均衡器抽头较少，则整个算法的计算得到化简。

通过以上分析可知，以上方法不仅对权向量进行了稀疏处理，而且对互四阶累积量的计算也采用了稀疏化处理。因此，该算法可称为稀疏互四阶累积量稀疏权超指数算法，记为 SSSE(sparse mutual four order cumulant sparse weight super exponential algorithm)。下面通过仿真考察该算法的性能。

仿真采用文献[12]的信道，是一条典型的稀疏水声信道，其传递函数为

$$h(z) = 1 + 0.7z^{-17}$$

该信道第二显著抽头距离第一显著抽头有 17 个码间间隔，具有明显稀疏性。采用 BPSK 信号，用于估计的数据长度为 2000 点，权向量长度为 81，信噪比 15dB。式(1.2.67)中的参数 $\delta_f$ 取 19。用于舍弃较小的 $\boldsymbol{d}$ 向量元素和权向量抽头的门限均取 0.1。200 次蒙特卡罗仿真的均方误差曲线如图 1.12 所示。同时，作为对比，也给出了 SE 算法的收敛曲线。图 1.12

表明，SE 算法和 SSSE 算法经历 5 次迭代都收敛了。由式(1.2.70)的证明，如果信道具有稀疏性，则在第一次迭代中，$\boldsymbol{d}$ 向量显著元素即呈现稀疏性，且显著元素间隔即稀疏信道显著抽头间隔。仿真中，第一次迭代求取了 $\boldsymbol{d}$ 向量的所有元素，如图 1.13(a)所示。可见 $\boldsymbol{d}$ 向量的显著元素只有 3 个，具有明显的稀疏性，且显著元素间隔正是稀疏信道显著抽头间隔 17，符合式(1.2.70)给出的规律。经门限判决后，第 2 次迭代仅计算了 $\boldsymbol{d}$ 向量 3 个显著元素；第 3～5 次迭代中仅计算了 2 个显著元素，如图 1.13(b)所示。而权向量在 5 次迭代中均只有 4 个较显著抽头参与迭代，如图 1.13(c)所示。

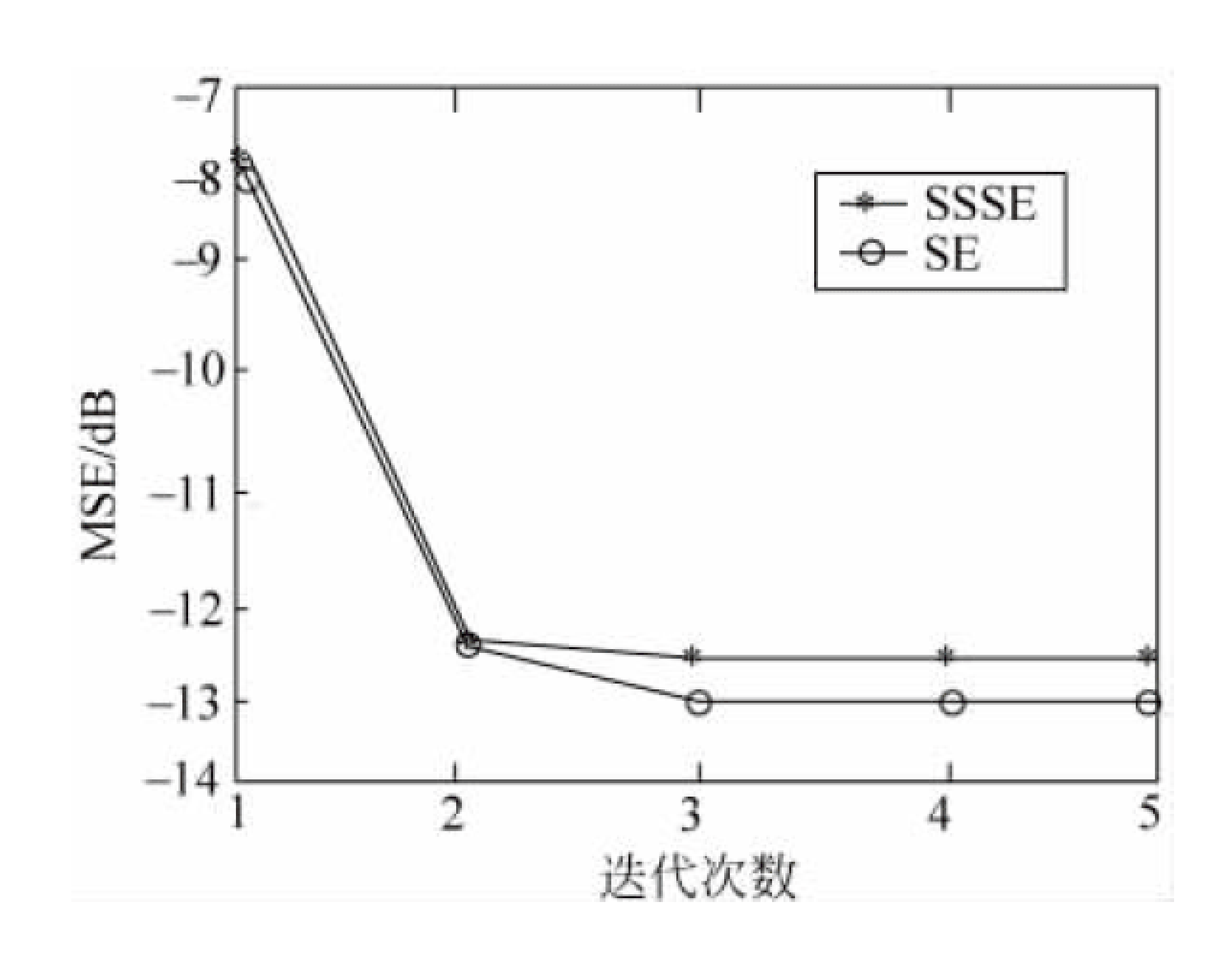
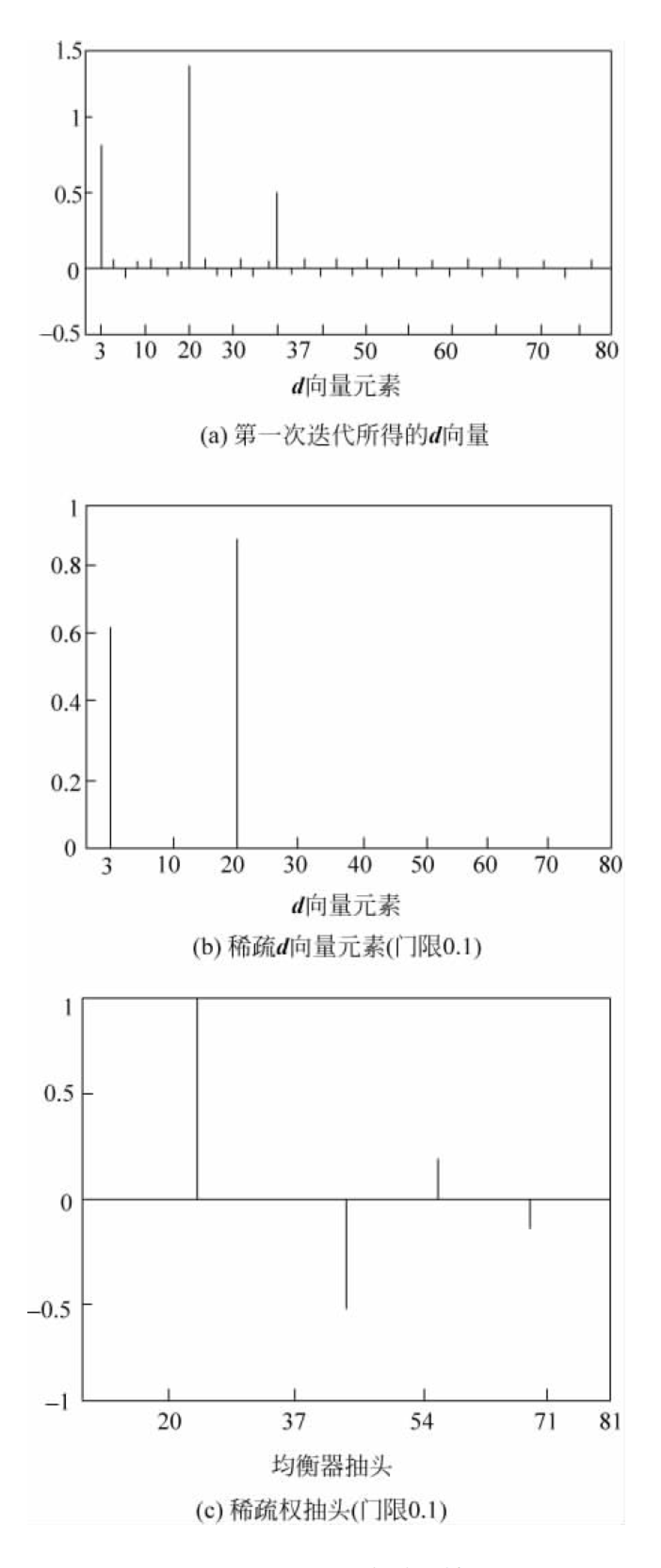



图 1.12　SSSE 算法与 SE 算法收敛均方误差曲线

下面对比 SSSE 算法与 SE 算法的计算量。当然，当权向量长度较长时，式(1.2.64)和式(1.2.65)的计算量不能忽略，但这是对 SSSE 算法和 SE 算法都具有的计算量，在二者比较时暂不考虑。主要通过式(1.2.69)观察两算法计算量的差别。对于 SSSE 算法，用 $M$ 表示数据点数，第一次迭代中 $\boldsymbol{d}$ 向量所有元素都被计算，则计算量为$(2\times81+1)M$，其余四次迭代中按 3 个元素被计算，则计算量为 $4\times(2\times3+1)M$；在 5 次迭代中，参与迭代的权向量抽头为 4，所以恢复均衡器输出的计算量为 $5\times4M$；加上估计相关矩阵的 $81M$ 次计算，总共为 $292M$；而相应的 SE 算法的计算量为 $1301M$，可见，SSSE 算法大大简化了计算量。由图 1.13 所示的均方误差曲



1.5
1
0.5
0
−0.5
3 10 20 30 37 50 60 70 80
d向量元素
(a) 第一次迭代所得的d向量
1
0.8
0.6
0.4
0.2
0
3 10 20 30 40 50 60 70 80
d向量元素
(b) 稀疏d向量元素(门限0.1)
1
0.5
0
−0.5
−1
20 37 54 71 81
均衡器抽头
(c) 稀疏权抽头(门限0.1)


图 1.13 明显的稀疏性

线可见，SSSE 算法的剩余均方误差比 SE 算法增大了 0.4dB，仅略有增加。因此，在实际应用中，应适当选择门限，控制 SSSE 算法稀疏化的程度，在减小算法计算量的同时，保证误码率满足系统指标要求。

## 1.3 基于正交小波变换的自适应均衡理论

均衡器输入信号自相关矩阵特征值的分散程度 $\lambda_{max}/\lambda_{min}$ 是影响最小均方(LMS)自适应算法收敛速度的主要因素($\lambda_{max}$ 和 $\lambda_{min}$ 分别为输入信号自相关矩阵的最大、最小特征值)。也就是说，均衡器收敛速度与输入信号自相关矩阵的最大、最小特征值比值有关，且 $\lambda_{max}/\lambda_{min}$ 越大，收敛速度越慢，否则，正好相反[13,14]。

为了提高 LMS 算法的收敛速度，研究人员通过对信号进行归一化正交小波变换，能使其自相关阵接近对角阵，即降低输入信号的自相关，这在一定程度上能够加快收敛速度。

本节将小波变换引入自适应均衡器中，研究了基于正交小波变换的自适应均衡算法，希望该算法获得更好的收敛性能。

### 1.3.1 均衡器的正交小波表示[15-19]

根据 Mallat 塔形算法思想，在有限尺度下，有限冲击响应均衡器 $f(k)$ 可由一族正交小波函数及尺度函数来表示，即

$$f(k) = \sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{j,n} \cdot \varphi_{j,n}(k) + \sum_{n=0}^{k_J} v_{J,n} \cdot \phi_{J,n}(k) \tag{1.3.1}$$

式中，$k=0,1,\cdots,L-1$，$L$ 为均衡器长度；$J$ 为最大尺度；$k_j=L/2^j-1(j=1,2,\cdots,J)$为尺度 $j$ 下小波函数的最大平移。其中，$d_{j,n}$和 $v_{J,n}$分别为

$$\begin{cases} d_{j,n} = \langle f(k), \varphi_{j,n}(k) \rangle \\ v_{J,n} = \langle f(k), \phi_{J,n}(k) \rangle \end{cases} \tag{1.3.2}$$

由于 $f(k)$的特性由 $d_{j,n}$和 $v_{J,n}$反映出来，故称 $d_{j,n}$和 $v_{J,n}$为均衡器的权系数。根据信号传输理论，均衡器输出为

$$
\begin{aligned}
z(k) &= \sum_{i=0}^{L-1} f_i(k) \cdot y(k-i) \\
&= \sum_{i=0}^{L-1} y(k-i)\left[\sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{j,n}(k)\cdot \varphi_{j,n}(i) + \sum_{n=0}^{k_J} v_{J,n}(k)\cdot \phi_{J,n}(i)\right] \\
&= \sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{j,n}(k)\left[\sum_{i=0}^{L-1} y(k-i)\cdot \varphi_{j,n}(i)\right] + \\
&\quad \sum_{n=0}^{k_J} v_{J,n}(k)\left[\sum_{i=0}^{L-1} y(k-i)\cdot \phi_{J,n}(i)\right] \\
&= \sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{j,n}(k)\cdot u_{j,n}(k) + \sum_{n=0}^{k_J} v_{J,n}(k)\cdot s_{J,n}(k)
\end{aligned} \tag{1.3.3}
$$

式中

$$
\begin{cases}
u_{j,n}(k) = \sum_i y(k-i)\varphi_{j,n}(i) \\
s_{J,n}(k) = \sum_i y(k-i)\phi_{J,n}(i)
\end{cases} \tag{1.3.4}
$$

式(1.3.3)实质上相当于对输入 $y(k)$作离散正交小波变换，$u_{j,n}(k)$为尺度为 $j$ 平移为 $n$ 的小波变换系数，$s_{J,n}(k)$为尺度为 $J$ 平移为 $n$ 的尺度变换系数。而式(1.3.3)表明，均衡器 $k$ 时刻输出 $z(k)$等于输入 $y(k)$经小波变换后的相应变换系数 $u_{j,n}(k)$和 $s_{J,n}(k)$与均衡器系数 $d_{j,n}$和 $v_{J,n}$的加权和。也就是说，将小波引入均衡器实质是对输入信号进行正交变换，从而改变了均衡器结构。

由式(1.3.1)知，小波系数 $u_{j,n}(k)$与尺度系数 $s_{J,n}(k)$的值依赖于小波函数 $\varphi(k)$与尺度函数 $\phi(k)$，而在实际中除了 Harr 小波外，小波函数 $\varphi(k)$与尺度函数 $\phi(k)$并没有明确的表达式，因而利用 Mallat 算法能够解决这一问题。

由 Mallat 塔形分解算法，对于长度为 $L$ 的离散信号 $\boldsymbol{S}_0=[s_{0,1}\ s_{0,2}\cdots s_{0,L-1}]^{\mathrm{T}}$，则其分解结构如图 1.14 所示。

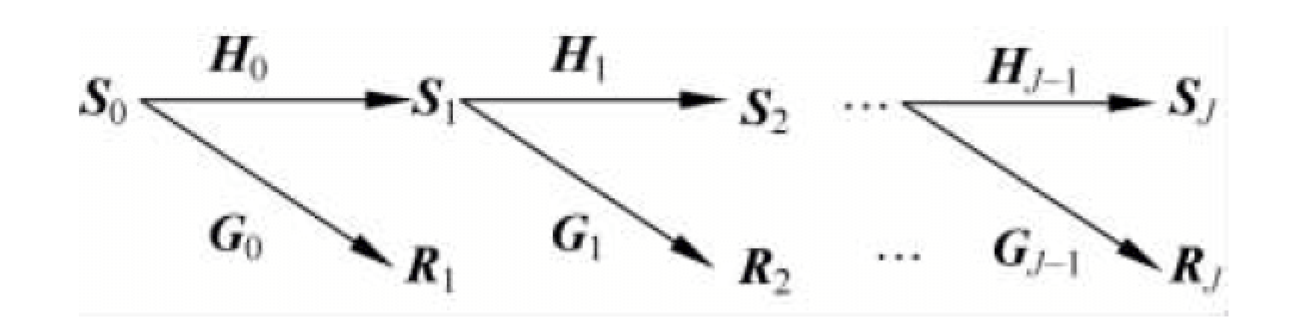



图 1.14 Mallat 分解结构

图 1.14 中，$\boldsymbol{S}_j=[s_{j,0}\quad s_{j,2}\quad \cdots\quad s_{j,k_j}]^{\mathrm{T}}$，$\boldsymbol{R}_j=[u_{j,0}\quad u_{j,2}\quad \cdots\quad u_{j,k_j}]^{\mathrm{T}}$，$k_j$ 定义同上，$H_j$ 和 $G_j$ 分别为由小波滤波器系数 $h(k)$ 和尺度滤波器系数 $g(k)$ 所构成的矩阵，且 $H_j$ 和 $G_j$ 中每个元素分别为 $H_j(l,k)=h(k-2l)$，$G_j(l,k)=g(k-2l)$，$(l=1\sim L/2^{j+1},k=1\sim L/2^j)$。若令 $\boldsymbol{R}=[\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_J,\boldsymbol{S}_J]^{\mathrm{T}}$，$\boldsymbol{V}=[\boldsymbol{G}_0;\ \boldsymbol{G}_1\boldsymbol{H}_0;\ \boldsymbol{G}_2\boldsymbol{H}_1\boldsymbol{H}_0;\ \boldsymbol{G}_{J-1}\boldsymbol{G}_{J-2}\cdots\boldsymbol{H}_1\boldsymbol{H}_0;\ \boldsymbol{H}_{J-1}\boldsymbol{H}_{J-2}\cdots\boldsymbol{H}_1\boldsymbol{H}_0]$，$\boldsymbol{V}$ 为正交小波变换矩阵且为一正交阵($\boldsymbol{V}\cdot\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{I}$)，且有

$$\boldsymbol{R}=\boldsymbol{V}\boldsymbol{S}_0 \tag{1.3.5}$$

这样即实现了信号的正交小波变换。也就是说要求 $\boldsymbol{R}$ 的值，只需知道 $h(k)$ 和 $g(k)$ 的值即可(通常实际中 $h(k)$ 和 $g(k)$ 的值都是已知的)，而不必再求 $\varphi_{j,n}(k)$ 与 $\phi_{J,n}(k)$。

若令式(1.3.5)中

$$\boldsymbol{S}_0=\boldsymbol{y}(k)=[y(k)\quad y(k-1)\quad \cdots\quad y(k-L+1)]^{\mathrm{T}}$$

$$\boldsymbol{R}(k)=[u_{1,0}(k),r_{1,1}(k),\cdots,u_{J,k_J}(k),s_{J,0}(k),\cdots,s_{J,k_J}(k)]^{\mathrm{T}}$$

且均衡器未知权系数

$$\boldsymbol{f}(k)=[d_{1,0}(k),d_{1,1}(k),\cdots,d_{J,k_J}(k),v_{J,0}(k),\cdots,v_{J,k_J}(k)]^{\mathrm{T}}$$

则可以得到引入正交小波变换的自适应均衡器结构，如图 1.15 所示。

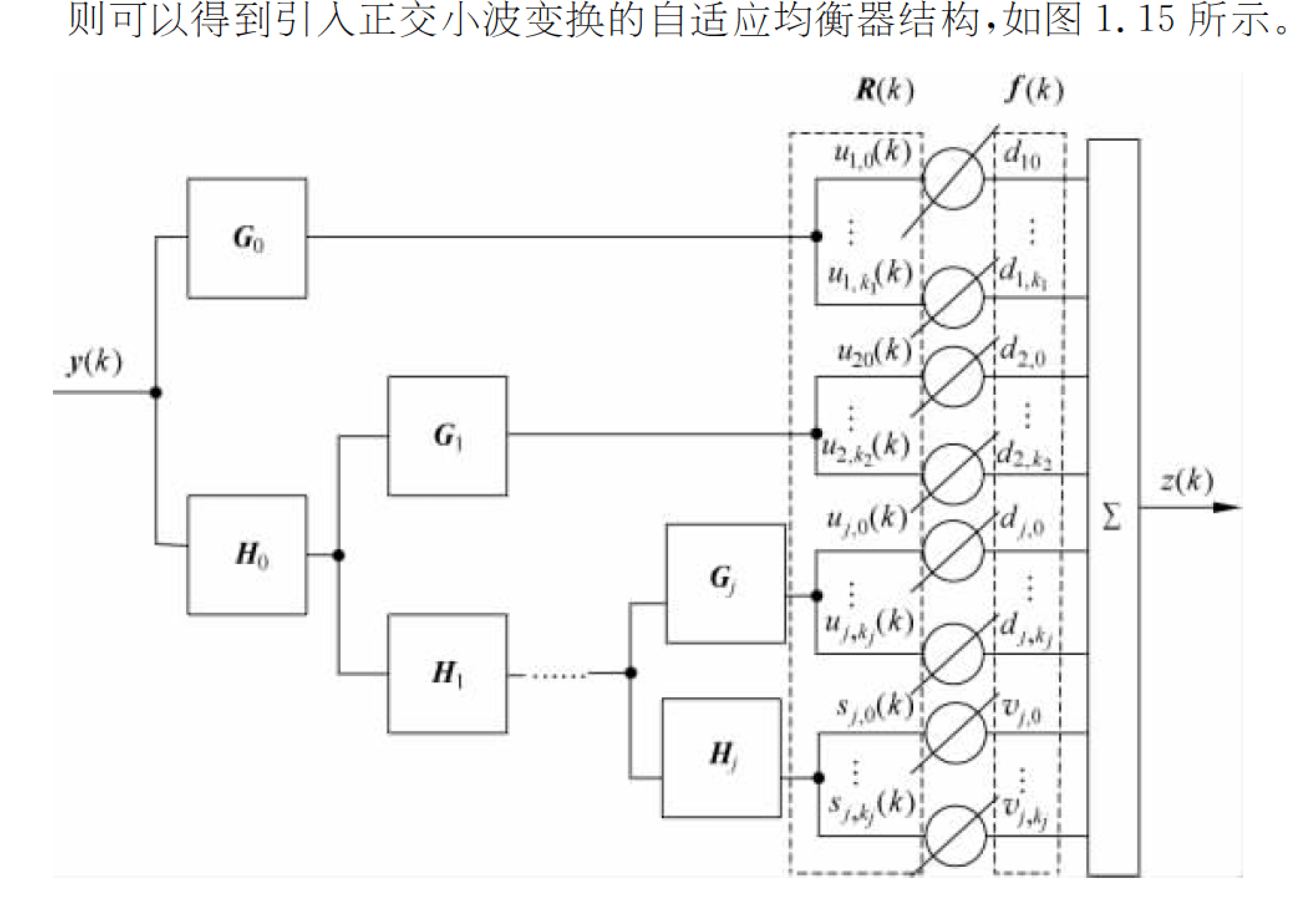



图 1.15　基于正交小波变换的自适应均衡器结构

图 1.15 可以简化为图 1.16。

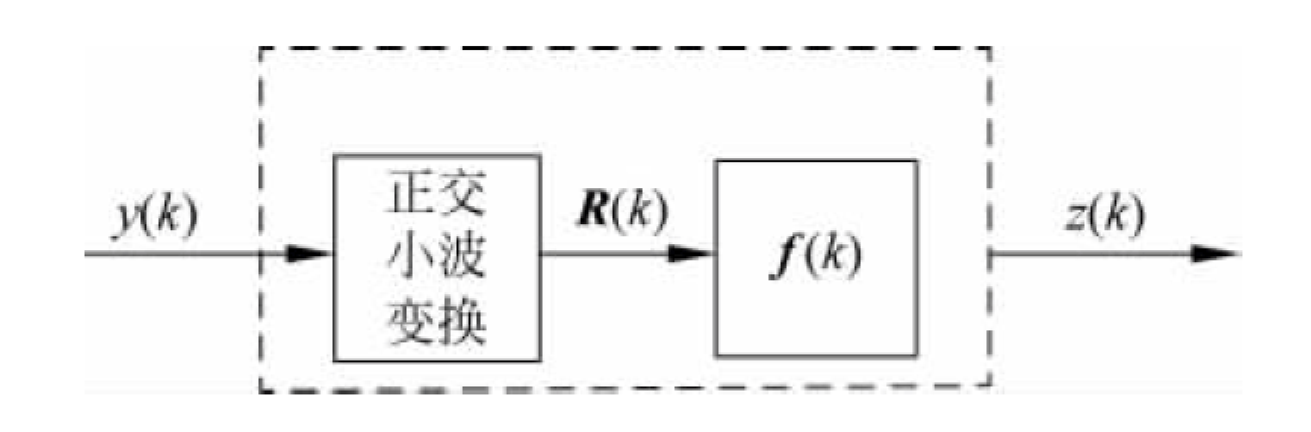



图 1.16　图 1.15 的简化

## 1.3.2　基于正交小波变换的 LMS 均衡算法

### 1. 算法原理

图 1.17 中，如果采用 LMS 算法更新权向量，则构成基于正交小波变换的自适应均衡算法。

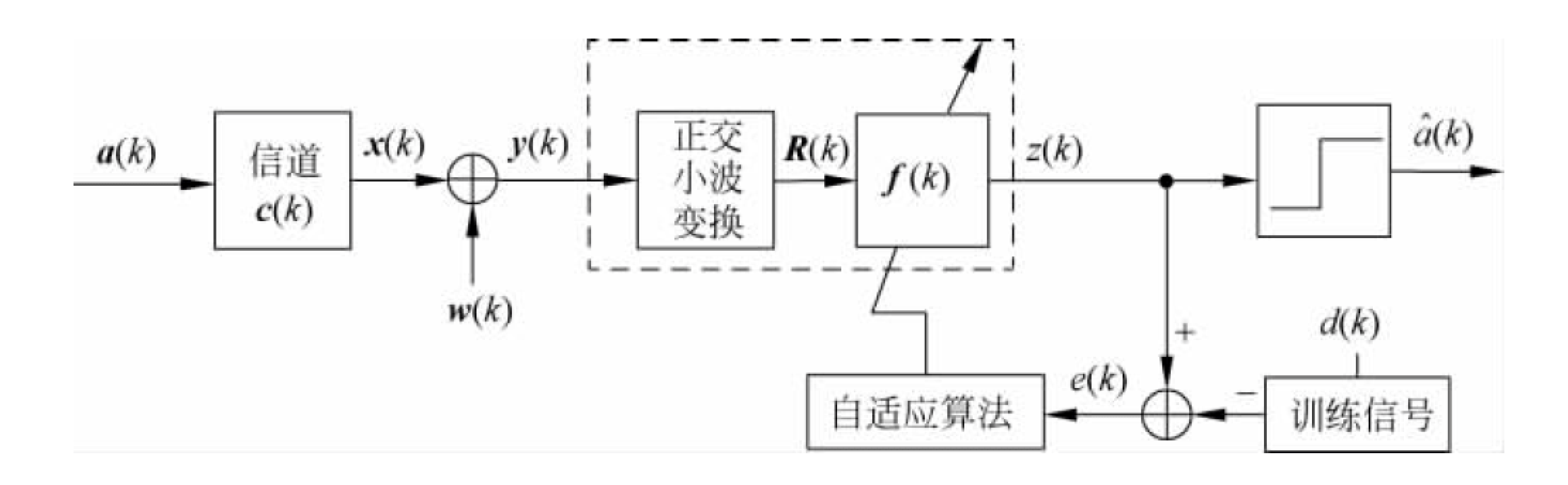



图 1.17　引入正交小波变换的自适应均衡原理

由图 1.17 知，经过小波变换后均衡器输出

$$\boldsymbol{R}(k)=\boldsymbol{V}\boldsymbol{y}(k) \tag{1.3.6}$$

在最小均方误差准则下，代价函数为

$$J(k)=E[e^2(k)] \tag{1.3.7}$$

式中

$$e(k)=z(k)-d(k) \tag{1.3.8}$$

$$z(k)=\boldsymbol{R}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{1.3.9}$$

这时，均衡器权向量迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\frac{\mu}{2}\frac{\partial J(k)}{\partial \boldsymbol{f}(k)} \tag{1.3.10}$$

式中，$\mu$ 为迭代步长，$\frac{\partial J(k)}{\partial \boldsymbol{f}(k)}$为代价函数对权向量的梯度，而

$$\frac{\partial J(k)}{\partial \boldsymbol{f}(k)} = E\left[2e(k)\frac{\partial e(k)}{\partial \boldsymbol{f}(k)}\right] = E\left[2e(k)\frac{\partial[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{R}(k) - d(k)]}{\partial \boldsymbol{f}(k)}\right]$$
$$= E[2e(k)\boldsymbol{R}(k)]$$

对$\frac{\partial J(k)}{\partial \boldsymbol{f}(k)}$取瞬时值后，代入式(1.3.10)

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu e(k)\boldsymbol{R}(k) \tag{1.3.11}$$

由于在同一尺度下，对不同的平移 $k$，$r_{j,n}(k)$间的相关性很小，$s_{J,n}(k)$间的相关性也很小，因此，可对变换后的信号能量作归一化处理，这时式(1.3.11)修改为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu \hat{\boldsymbol{R}}^{-1}(k)e(k)\boldsymbol{R}(k) \tag{1.3.12}$$

式中，$\hat{\boldsymbol{R}}^{-1}(k) = \mathrm{diag}[\sigma_{j,0}^2(k), \sigma_{j,1}^2(k), \cdots, \sigma_{J,k_j}^2(k), \sigma_{J+1,0}^2(k), \cdots, \sigma_{J+1,k_j}^2(k)]$，$\sigma_{j,k}^2(k)$与 $\sigma_{J+1,k_j}^2(k)$分别表示对 $u_{j,n}(k)$与 $s_{J,n}(k)$的平均功率估计，其递推估计公式为

$$\hat{\sigma}_{j,n}^2(k+1) = \beta_\sigma \hat{\sigma}_{j,n}^2(k) + (1-\beta_\sigma)\,|\,u_{j,n}(k)\,|^2 \tag{1.3.13}$$

$$\hat{\sigma}_{J+1,n}^2(k+1) = \beta_\sigma \hat{\sigma}_{J+1,n}^2(k) + (1-\beta_\sigma)\,|\,s_{J,n}(k)\,|^2 \tag{1.3.14}$$

式中，$\beta_\sigma$ 为平滑因子，且 $0<\beta_\sigma<1$，一般取 $\beta_\sigma$ 值比较接近于 1，$R^2 = E[|a(k)|^4]/E[|a(k)|^2]$。

2. 性能分析

由前面分析可知，LMS 算法的收敛速度取决于输入信号自相关矩阵最大、最小特征值比值，即矩阵 $\boldsymbol{R}$ 的条件数

$$\mathrm{cond}(\boldsymbol{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{1.3.15}$$

该值越小，收敛越快，因而引入小波变换可以加快收敛速度。下面我们从理论上来给予证明。

设输入信号为实信号 $\boldsymbol{y}(k)$，其输入自相关矩阵为 $\boldsymbol{R}_{yy}$；设信号经小波

变换后的自相关矩阵设为 $\boldsymbol{R}_{RR}$，则 $\boldsymbol{R}_{yy}$ 与 $\boldsymbol{R}_{RR}$ 均为实对称矩阵，因而存在正交阵 $\boldsymbol{Q}_y$ 和 $\boldsymbol{Q}_R$，使得

$$\boldsymbol{R}_{yy} = \boldsymbol{Q}_y \boldsymbol{\Lambda}_y \boldsymbol{Q}_y^{-1} \tag{1.3.16}$$

$$\boldsymbol{R}_{RR} = \boldsymbol{Q}_R \boldsymbol{\Lambda}_R \boldsymbol{Q}_R^{-1} \tag{1.3.17}$$

式中，$\boldsymbol{\Lambda}_y$ 和 $\boldsymbol{\Lambda}_R$ 分别是 $\boldsymbol{R}_{yy}$ 和 $\boldsymbol{R}_{RR}$ 的特征值对角阵，即

$$\boldsymbol{\Lambda}_y = \mathrm{diag}[\lambda_1^y \quad \lambda_2^y \quad \cdots \quad \lambda_N^y] \tag{1.3.18}$$

$$\boldsymbol{\Lambda}_R = \mathrm{diag}[\lambda_1^R \quad \lambda_2^R \quad \cdots \quad \lambda_N^R] \tag{1.3.19}$$

且其特征值均为正数。这时

$$\begin{aligned}\boldsymbol{R}_{RR} &= \boldsymbol{Q}_R \boldsymbol{\Lambda}_R \boldsymbol{Q}_R^{-1} = E[\boldsymbol{R}(h)\boldsymbol{R}^{\mathrm{T}}(k)] = E\{\boldsymbol{V}\boldsymbol{y}(k)[\boldsymbol{V}\boldsymbol{y}(k)]^{\mathrm{T}}\} \\ &= E\{\boldsymbol{V}\boldsymbol{y}(k)\boldsymbol{y}(k)^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\} = \boldsymbol{V}\boldsymbol{R}_{yy}\boldsymbol{V}^{\mathrm{T}}\end{aligned} \tag{1.3.20}$$

又由式(1.3.16)和式(1.3.17)知

$$\boldsymbol{R}_{RR} = \boldsymbol{Q}_R \boldsymbol{\Lambda}_R \boldsymbol{Q}_R^{-1} = \boldsymbol{V}\boldsymbol{Q}_y \boldsymbol{\Lambda}_y \boldsymbol{Q}_y^{-1}\boldsymbol{V}^{\mathrm{T}} \tag{1.3.21}$$

所以有

$$\boldsymbol{\Lambda}_R = \boldsymbol{Q}_R^{\mathrm{T}}\boldsymbol{V}\boldsymbol{Q}_y \boldsymbol{\Lambda}_y \boldsymbol{Q}_y^{-1}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{Q}_R = \boldsymbol{P}\boldsymbol{\Lambda}_y \boldsymbol{P}^{\mathrm{T}} \tag{1.3.22}$$

式中，$\boldsymbol{P}=\boldsymbol{Q}_R^{\mathrm{T}}\boldsymbol{V}\boldsymbol{Q}_y$，而由式(1.3.22)可得

$$\lambda_n^R = \sum_{i=1}^{M} p_{ni}^2 \lambda_n^y, \quad n = 1,2,\cdots,M \tag{1.3.23}$$

式中，$p_{ki}$ 为矩阵 $\boldsymbol{P}$ 中的第 $(n,i)$ 个元素，又因各特征值均为正数，故由式(1.3.23)得

$$0 < \lambda_{\min}^y \min_n \left( \sum_{i=1}^{M} p_{ni}^2 \right) \leqslant \lambda_{\min}^R \leqslant \lambda_{\max}^R \leqslant \lambda_{\max}^R \max \left( \sum_{i=1}^{M} p_{ni}^2 \right) \tag{1.3.24}$$

对于一般情况下，有 $\min\limits_n \left( \sum\limits_{i=1}^{N} p_{ni}^2 \right) \approx \max\limits_n \left( \sum\limits_{i=1}^{N} p_{ni}^2 \right)$。这样，式(1.3.24)意味着

$$\frac{\lambda_{\max}^R}{\lambda_{\min}^R} \leqslant \frac{\lambda_{\max}^y}{\lambda_{\min}^y} \tag{1.3.25}$$

由此可见，经小波变换后矩阵 $\boldsymbol{R}_{RR}$ 的最大、最小特征值之比小于 $\boldsymbol{R}_{yy}$ 的最大最小特征值之比，即引入小波变换后，性能得到改善。

# 参 考 文 献

[1] Johnson C R, Schniter P, Enders T J. Blind equalization using the constant modulus criterion: a review[J]. Proceedings of the IEEE, 1998, 86(10): 1927-1950.

[2] Heinz Mathis, Scott C Douglas. Bussgang blind deconvolution for impulsive signals[J]. IEEE Transaction on Signal Processing,2003,51(7): 1905-1916.

[3] Gianpiero Panci, Stefania Colonnese, Patrizio Campisi, et al. Blind equalization for correlated input symbols: a bussgang approach[J]. IEEE Transaction on Signal Processing,2005,53(5): 1860-1870.

[4] Shafayat Abrar, Azzedine Zerguine, Maamar Bettayeb. Recursive least-squares backpropagation algorithm for Stop-and-Go decision-directed blind equalization [J]. IEEE Transaction on Neural Network,2002,13(6): 1472-1481.

[5] Zhi Ding, Rodney A Kennedy, Brian D O Anderson, et al. Local convergence of the Sato blind equalizer and generalizations under practical constraints[J]. IEEE Transaction on Information Theory,1993,39(1): 129-145.

[6] Picchi G, Prati G. Blind equalization and carrier recovery using a "stop-and-go" decision directed algorithm[J]. IEEE Transactions on Communication,1987,35(9): 877-887.

[7] 郭业才. 通信信道盲均衡算法与仿真[M]. 北京: 气象出版社,2009.

[8] 王峰. 基于阶统计量的水声信道均衡理论与算法[D]. 西安: 西北工业大学,2003.

[9] 郭业才. 自适应盲均衡技术[M]. 合肥: 合肥工业大学出版社,2007.

[10] 郭业才. 通信信号分析与处理[M]. 合肥: 合肥工业大学出版社,2009.

[11] O Shalvi, E Weinstein. Super-exponential methods for blind equalization[J]. IEEE Transaction on Information Theoty,1993,39(2): 505-519.

[12] B Geller, V Capellano, G Jourdain. Equalizer for real time high rate Transmission in underwater communications[C]. International conference on Acoustics, Speech, and Signal Procession,1995,5: 3179-3182.

[13] M Doroslovacki, H Fan. Wavelet-based linear system modeling and adaptive filtering[J]. IEEE Trans. on Signal Processing,1996,44(5): 1156-1167.

[14] T Cooklev. An efficient architecture for orthogonal wavelet transforms[J].

IEEE Signal Processing Letters,2006,13(2):77-79.

[15] P S Long. Dimensional finite wavelet Filters[J]. Journal of Computation Mathematics,2003,5:595-602.

[16] 廖娟,郭业才,刘振兴,等.基于遗传优化的正交小波分数间隔盲均衡算法[J].兵工学报,2011,32(3):268-273.

[17] S Attallah. The wavelet transform-domains LMS adaptive filter with partial subband coefficient updating[J]. IEEE Trans Circuits and Systems,2006,53(1):8-12.

[18] Sun Jing, Guo Yecai. Orthogonal wavelet transform sign decision dual-mode fixed-weight multi-modulus blind equalization algorithm[C]. 2011 International Conference on Aerospace Engineering and Information Technology,AEIT2011, May 5-6,2011,Beijing,China,2011:78-83.

[19] 郭业才.模糊小波神经网络盲均衡理论、算法及实现[M].北京:科学出版社,2011.

# 第 2 章　基于遗传算法优化的盲均衡算法

**【内容导语】** 本章在简要阐述遗传算法的概念、用语、基本操作(编码、种群初始化、确定适应度函数、执行交叉、变异、选择等)后,将遗传算法、自适应遗传算法、混合模拟退火遗传算法和改进的混合遗传算法引入到常模盲均衡、正交小波盲均衡、正交小波超指数迭代盲均衡等算法中,优化了盲均衡算法的性能。

近年来,随着计算科学的发展以及人们对自然界认识的深入,提出了一些用于解决优化问题的生物启发式算法,如遗传算法、粒子群算法、人工鱼群算法、果蝇算法、模拟退火算法等。这些启发式算法不需要依赖梯度计算,能较好地找到全局最优解。与传统的数学规划法和准则法相比,启发式算法的主要优势有:①在一般情况下,算法能否收敛到全局最优解与初始群体无关;②全局搜索能力强;③使用范围广,能有效求解不同类型的问题。因此,启发式智能优化算法及其应用是值得研究的课题。

## 2.1 遗传算法

遗传算法是模拟自然进化过程而形成的一种搜索最优解的随机搜索法，它最早是由美国密歇根大学的 Holland 教授提出来的，其主要特点是直接对求解对象进行操作，不需要所求的目标函数满足求导和函数连续性条件，它采用适者生存的进化规律求解问题的最优解[1-4]。遗传算法的研究对象是一个由基因编码的一定数目的个体组成的种群，每个个体其实是染色体带有特征的实体，各个个体代表着需求解的问题的可能解。染色体作为生物体遗传物质的主要载体，是多个基因的组合。它有两种表现模式：基因型和表现型。在遗传算法中，一开始就需要实现从表现型到基因型的映射，即编码工作。为了降低编码工作中的复杂程度，一般用简化的编码方法，最常用的就是二进制编码。初始种群设置好之后，按照生物进化机制，通过逐代进化得到越来越好的所求问题的近似解。对于遗传算法进化的每一代，遗传算法会用选择算子选出种群中适应能力强的个体，然后对这些个体进行组合交叉、变异操作，产生下一代种群，它代表着一个新的解集。遗传算法的选择、交叉和变异操作过程会使后生代种群在环境中的适应生存能力强于前一代种群，这样，末代种群中的适应度函数值（简称适应度值）最大的个体经过解码就可以视为求解问题的最优解近似值。

### 2.1.1 遗传算法的基本用语

遗传算法会使用一些与自然遗传有关的基础用语，而这些基础用语对于遗传算法研究和应用是很重要的。

1. 染色体、基因串或个体

细胞是生物体的基本结构和功能单位，细胞内的细胞核主要由一种微小的丝状染色体构成，而染色体是携带着基因信息的数据结构，是生物遗传

信息的主要载体,也可称为基因串或个体,一般用二进制位串来表示。

2. 基因、基因座与等位基因

基因是染色体的一个片段,它是控制生物性状的遗传物质的功能单位和结构单位,通常为单个参数的编码值,若干个基因组成一个染色体。每个基因对应的染色体中的位置称作基因座,每个基因所取的值称作等位基因。其中,染色体特征和生物个体性状是由基因和基因座决定的。

3. 基因型和表现型

染色体可用基因型和表现型这两种相应的模式表示。所谓基因型是指个体的内部表现,它是由该个体特有的基因组成,该基因与个体的表现型有着密切的联系;而表现型是个体在生存环境中所具有的外部表现,它是个体基因型与生存环境条件相互作用的结果,即在不同的生存环境条件下,同一种基因型的生物个体可以有着不同的表现型。

4. 编码操作与译码操作

在执行遗传算法时,操作过程中存在基因型和表现型两者之间相互转化的操作,将表现型转换为基因型称为编码操作,它实现所求解问题的搜索空间中的参数与遗传空间中个体的转换;另一个是译码操作,它将基因型转换为表现型,实现遗传空间中的个体与搜索空间中的参数或解的转换。

5. 群体规模

在标准的遗传算法中,一定数量的个体组成种群,在种群中个体数目的大小称为种群的大小,即群体规模。

### 2.1.2 遗传算法的基础理论

1. 模式定理

遗传算法模拟生物个体之间的选择、交叉和变异等遗传操作来逐步搜

索问题的最优解。在这个搜索的过程中,逐步产生的每一代个体在它的编码串组成结构上与其父代个体之间有一些相似的结构联系。若将个体之间具有相似结构特点的编码串与某些相似模板相联系,则遗传算法中对个体的搜索就是对这些相似模板的搜索,这样,就需要引入模式的概念。模式表示种群中的个体基因串中某些特征位相同的结构,它描述了个体编码串中具有相似结构特征的一个串子集。例如,在二进制编码串中,用 0 和 1 这两个元素所组成的一个编码串来表示个体,用 0,1 和“ * ”这三个元素所组成的一个编码串来表示模式,其中“ * ”表示任意字符,既可以为“1”,也可以为“0”,可见,一个模式可以隐含在多个编码串中,不同的编码串之间通过模式相互联系,使遗传算法中的编码串的运算转化为模式的运算。为了定量地估计模式运算,将模式中具有确定基因值的位置个数称为该模式中的阶,如果阶数越高,与该模式匹配的样本数就越少,确定性就越高;将模式的定义长度表示为模式中有确定基因值的基因位从第一个到最后一个之间的距离。因此,在对遗传算法的模式概念进行相关分析后,可以将遗传算法看作是一种模式的运算,即一个模式中的各个样本经过相关遗传操作进化成一些新的样本和新的模式,并且,那些低阶、短定义长度且平均适应度值高于群体平均适应度值的模式,经过选择、交叉和变异操作后,将成指数增长,这些正是模式定理[4-7]所揭示的内容。

从上面分析可知,模式定理在一定程度上证明了遗传算法的有效性,但是它仍存在一些缺陷:

- 模式定理只适合二进制编码,没有适合其他编码方案的相关结论。
- 模式定理只提供本代包含某个模式的个体数的下限,并不能根据此下限来确定算法的收敛与否。
- 模式定理没有解决遗传算法设计中控制参数的选取等问题。

2. 积木块假设

遗传算法是一个随机概率搜索的过程,而不是对搜索空间中每个基因都进行检测和遗传操作,只是将一些较好的模式像堆积木一样拼接在一起,

从而使进化出的个体编码串在生存环境中的适应能力越来越强，而这些模式是低阶、短定义长度且平均适应度值高于群体平均适应度值的模式，它们经过选择、交叉和变异操作后，其样本将成指数增长，将符合这些条件的一些模式称为积木块。

模式定理指明了积木块的样本呈指数增长和用遗传算法寻找最优解的可能性，但是没有说明遗传算法一定能够搜索到最优解；而积木块假设能说明遗传算法一定能搜索到最优解。所谓积木块假设就是积木块经过选择、交叉和变异操作后，它们能够相互结合并形成适应度值更大的个体编码串，最后趋近全局最优解。它说明了用遗传算法求解各类问题的基本思想，在许多领域得到了应用，同时也证明了积木块假设的有效性。

3. 隐含并行性

由前面的讨论知，遗传算法的一个编码串中实际上隐含有多种不同的模式，所以遗传算法的实质是模式的运算。在二进制编码串中，如果编码串长度为 $l$，则该编码串中会隐含有 $2^l$ 种模式，在种群规模为 $N$ 的群体中就有 $2^l \sim N \cdot 2^l$ 种不同模式。随着这些模式的逐代进化，这些模式中的一些较长的定义长度的模式将被破坏，而较短的定义长度的模式将被保存下来。由此可见，遗传算法在运行过程中，每代除了处理 $N$ 个个体之外，还并行处理了与 $N$ 的立方成正比的模式数，此处的并行处理过程与一般的并行算法的处理过程不一样，它是一种隐含并行性，包含在处理过程内部。

### 2.1.3 遗传算法的基本操作

1. 编码

遗传算法直接处理的对象不是所求问题中的实际决策变量，而是用某方法将所求问题的可行解转化为遗传算法的编码个体再进行有关的遗传

操作，逐代搜索出适应度值较大的个体，并逐渐增加其在种群中数量，最后寻找出问题最优解的最佳近似解[8]。而遗传算法中能将所求问题的可行解转化为遗传算法能搜索个体的方法是编码，编码将求解问题的可行解从其解空间转换到遗传算法所能处理的搜索空间，它是遗传算法中的一个关键步骤。编码方法的选择与如何进行群体的遗传进化运算以及遗传进化运算的效率有着密切的关系，不同的编码方法会有不同的结果和计算效率。编码方法应满足三个规范条件：完备性、健全性和非冗余性[9]。针对一个具体的应用问题，如何设计一种编码方案是遗传算法的一个重要的研究方向，迄今为止人们已经提出了许多种不同的编码方案，主要有二进制编码、符号编码和浮点数编码这三种编码方案，可以根据不同的要求来选择编码方法。

（1）二进制编码

在遗传算法中，最常用的一种编码方案就是二进制编码。二进制编码的编码符号集是由一定数目的二进制符号 0 和 1 所组成，由二进制编码方法编码的个体基因型是一个二进制编码串。二进制编码方法的编解码操作简单、易于实现。

（2）符号编码

符号编码方法是指将一个无数值含义而只有代码含义的符号集，用来表示个体的每个基因值。符号编码方法不仅与积木块编码原则相符合，还便于遗传算法与其他相关的近似算法之间的结合使用。

（3）浮点数编码

浮点数编码是将个体的每个基因值用某一范围的一个浮点数来表示，个体的编码长度与所求问题决策变量的个数相对应。当遗传算法中个体基因值的表示范围较大时，浮点数编码优于其他的编码方法，也就是当搜索空间较大时，使用浮点数编码能很好地提高遗传算法的运算效率。

### 2. 初始种群

遗传算法是对一定数目个体组成的群体进行遗传操作，因此要给种群

赋初始值，它是遗传算法的起始搜索点，一般用随机方法产生。而一定数目的个体就构成了种群的规模，种群规模对遗传算法的性能也有一定的影响，种群的规模越大，种群的多样性就越丰富，减小了陷入局部收敛的可能性，但它会增加计算量；种群规模越小，则减小可行解在搜索空间的分布范围，增大了“早熟”收敛的可能性。所以，针对不同的问题要求，种群规模也会不同。

Goldberg[10]证明了使用二进制编码时，若个体串的长度为 $L$，则种群规模的最优值为 $2^{\frac{L}{2}}$。

### 3. 适应度函数

适应度函数就是度量个体适应度的函数。由于遗传算法主要是以个体的适应度值为参量来进行遗传操作的，基本不需要其他信息和适应度函数的连续可导性，因此遗传算法使用适应度值来衡量群体中各个个体在优化算法中有助于找到最优解的优良程度，个体的适应度值越大，则该个体被选择遗传到下一代的概率就越大，反之，被遗传到下一代的概率越小。因此，适应度函数的选择对于遗传算法搜索效率很重要。一般情况，适应度函数是由目标函数转换而来的，因为遗传算法的每步搜索信息仅通过所求问题的目标函数值就能得到，这样遗传算法在评价个体适应度值时能够体现所求问题的目标函数值的使用。评价个体适应度值的基本过程为：

- 通过解码将个体基因型转化为对应的表现型；
- 根据个体的表现型计算出该个体的目标函数值；
- 根据优化问题的类型，将目标函数值转化为个体的适应度值。

最优化问题一般可有两类，一类是求目标函数的最大值，另一类是求目标函数的最小值。设目标函数用 $J(x)$来表示，如果要求 $J(x)$的最小值，则将求最小值问题转化为求最大值问题，从而将函数值非负的适应度函数 $\mathrm{Fit}(x)$表示为

$$\mathrm{Fit}(x)=\frac{1}{J(x)} \tag{2.1.1}$$

反之，适应度函数为

$$\mathrm{Fit}(x)=J(x) \tag{2.1.2}$$

遗传算法中，适应度函数对遗传算法的收敛速度和能否搜索到最优解都有很大的影响。如果对每代的适应度值较大的个体强调过多，则会降低种群的多样性，使算法易陷入“早熟”的收敛现象；反之，使算法丢失适应度值较大的个体信息，不能达到合理的收敛。因此，为了提高遗传算法的有效性，适应度函数的设计需要满足下面一些条件：

- 适应度函数值必须是连续的非负单值，且有最大值；
- 适应度函数的设计应使适应度值对应解的优劣程度；
- 适应度函数的设计尽可能降低复杂度、减少计算量；
- 针对某类具体的问题，适应度函数设计应尽可能通用。

4. 选择算子

遗传算法使用选择算子来对群体中的个体进行优胜劣汰操作，即用选择算子来确定如何从上一代个体中选取个体遗传到下一代，该操作是以群体中各个个体的适应度评价为基础的，它能够避免某些基因的缺失、提高算法的全局收敛性。

常用的选择算子的操作方法如下。

(1) 比例选择算子

比例选择算子即轮盘选择，它是指各个个体被遗传到下一代的概率与该个体适应度值的大小成正比，即个体的适应度值越大，则它被选择遗传到下一代的概率越大。设种群规模大小为 $N$，个体 $i$ 的适应度为 $\mathrm{Fit}_i$，则个体 $i$ 被选择遗传的概率 $p_i$ 为

$$p_i=\frac{\mathrm{Fit}_i}{\sum_{i=1}^{N}\mathrm{Fit}_i} \tag{2.1.3}$$

（2）最优保存策略

在遗传算法进化过程中，优良个体会随着进化代数的增加而增多，但进化过程中选择、交叉、变异等操作是一些随机的遗传操作，它们有可能破坏当前群体中适应度值最大的个体，从而降低群体的平均适应度值，影响遗传算法的运行效率和收敛速度。为了降低适应度值最大的个体被破坏的可能性，在优胜劣汰操作中引入最优保存策略，即当前群体中适应度值最大的个体不参与交叉、变异运算，并且还替换本代群体中经过遗传操作后所产生的适应度值最小的个体。该方法能保证最优个体不被破坏，但它容易使某个局部最优个体不易被淘汰而降低算法的全局搜索能力。因此，该方法一般是与其他的一些选择操作方法联合使用的。

（3）排序选择方法

排序选择方法是由 Whitley 等提出的，它的主要着眼点是个体适应度值之间的大小关系，首先对群体中所有个体按其适应度值大小进行降序排序，然后根据具体求解问题，设计一个概率分配表，将各个概率值按上述排列次序分配给各个个体，各个个体将以所分配到的概率值作为其能够被遗传到下一代的概率。该方法是基于概率的选择操作，且选择概率是要事先确定的，所以会产生较大的选择概率。

（4）随机遍历抽样

随机遍历抽样法提供了零偏差和最小个体的扩展，首先设定需要选择的个体数目，然后用选择指针等距离的选择个体，其中选择指针的距离用需要选择的个体数目的倒数表示。

以上是几种常用的选择方法，它们对遗传算法性能的影响都不同，在实际应用时，根据所求问题的特点选择适合的选择方法。

### 5. 交叉算子

交叉运算是指两个相互配对的个体通过某种方式相互交换部分基因来形成两个新的个体，它是遗传算法区别于其他进化算法的重要特征，它在遗传算法中起着关键作用，它能使上一代群体中优良个体的特性有一定程

度的保持，同时，又是产生新个体的主要方法，对于增强遗传算法的全局搜索能力很重要。常用的交叉方法[7]如下。

(1) 单点交叉

单点交叉是最常用和最基本的交叉操作算子，它是指对群体中的个体进行两两配对，对每一对相互配对的个体随机设置一个交叉点，然后在该交叉点相互交换部分染色体。

(2) 多点交叉

多点交叉也被称为广义交叉，操作过程与单点交叉相似，不同的是它在个体编码串中随机设置了多个交叉点，但该交叉方法有可能破坏一些好的模式，甚至随着交叉点数的增加，个体结构被破坏的可能性也逐渐增大，使特性好的模式更容易破坏，影响遗传算法的性能，一般使用不多。

(3) 算术交叉

算术交叉是指两个个体进行线性组合并产生两个新的个体，它一般用于实数编码的求解问题中。

设算术交叉的两个个体为 $A_t$、$B_t$，则交叉产生的两个新个体 $A_{t+1}$、$B_{t+1}$为

$$A_{t+1} = \alpha B_t + (1-\alpha)A_t \tag{2.1.4}$$

$$B_{t+1} = \alpha A_t + (1-\alpha)B_t \tag{2.1.5}$$

式中，$\alpha$ 是一个比例因子。

除了上述的三种交叉方法外，还有均匀交叉、循环交叉，顺序交叉等。

6. 变异算子

变异运算是指将个体编码串中的某些基因变换为这些基因对应的等位基因，从而形成一个新的个体。变异运算改变的是个体编码串中的部分基因值，它是产生新个体的辅助方法，它与遗传算法的局部搜索能力相关联，与交叉算子相结合，实现遗传算法的全局搜索和局部搜索，提高了遗传算法的搜索性能。常用的变异算子如下。

(1) 基本位变异

基本位变异是指以变异概率随机指定个体编码串中某几位基因座,将这些基因座上的基因值转化为与其相应的等位基因值。该操作方法的对象只是个体编码串中的个别几个基因座上的基因值,且变异概率小,变异产生的作用不明显。

(2) 均匀变异

均匀变异是以变异概率将个体编码串中的每个基因座指定变异点,并将每一变异点原有的基因值转化为其对应的取值范围内一随机数。该方法使得算法能在整个搜索空间内自由搜索,能增加群体多样性,比较适合在算法的初始运行阶段,不适合于某一重点区域的局部搜索。

(3) 实值变异

一般情况下,较小的变异步长使变异操作较成功,但有时变异步长大又会加快优化速度。为了解决这个矛盾,引入实值变异,定义为

$$X' = X \pm 0.5\gamma\Delta \tag{2.1.6}$$

式中,$\gamma$ 为变量的取值范围;$X$ 为变异前变量取值;$X'$ 为变异后变量取值;$\Delta = \sum_{i=0}^{m-1} \frac{a(i)}{2^i}$,其中,$a(i)$取值为 1 或 0,且取 1 的概率为$\frac{1}{m}$,取 0 的概率为 $1-\frac{1}{m}$。

### 2.1.4 遗传算法的参数选择

遗传算法中的参数对算法性能有着重要影响。其中主要有以下几个参数[8]。

#### 1. 编码串长度

编码串长度与编码方法的选择有关。例如,二进制编码的编码串长度与所求问题要求的精度有关;而实数编码的编码串长度则等于决策变量的个数。

2．交叉概率

交叉操作对遗传算法中新个体的产生起着主要作用。交叉概率大，开辟新的搜索空间的能力强，但交叉概率过大，会破坏群体的优良模式，影响算法的优化性能；反之，交叉概率小，产生新个体的数量少，降低了群体多样性，可能使遗传算法的搜索陷入迟钝状态。一般建议取值范围是0.4～0.99。

3．变异概率

变异操作可以改善遗传算法局部搜索能力，保持群体多样性，防止早熟。变异概率大，产生新个体多，但变异概率过大则可能破坏群体的优良模式；反之，变异操作产生新个体少且抑制早熟能力差。一般建议取值范围为0.0001～0.1。

4．终止代数

终止代数作为遗传算法运行结束的一个条件，当遗传算法运行到预先设置好的代数之后就会停止运行，并将最后一代群体中的最佳个体作为所求问题的最优解输出。一般建议取值范围为100～1000。

### 2.1.5 遗传算法的特点

遗传算法是模拟生物的遗传和进化过程而形成的一种全局随机搜索算法。它与传统的优化算法存在几点不同之处[3]：

(1) 传统优化算法的优化对象是决策变量的实际值；而遗传算法的优化对象是决策变量的某种形式的编码，有利于遗传操作算子的应用，且遗传算法在一些没有数值或很难有数值概念的优化问题中更有其独特的优越性。

(2) 传统优化算法的目标函数应满足连续、可导等条件；而遗传算法

只使用由目标函数变换而得的适应度值来评估基因个体,并在此基础上进行相应的遗传操作。遗传算法对搜索空间没有任何特殊要求,其适应度函数不受连续可微条件的约束,对适应度函数的唯一要求是遗传算法的编码必须与所求问题的可行解空间相对应。

(3) 传统优化算法搜索最优解的迭代过程是从解空间中的一个初始点开始;而遗传算法具有隐含的并行性和并行计算能力,它对最优解的搜索是从一个初始群体开始,这个初始群体是由很多个体组成的,这样搜索点多,提供的搜索信息多,从而提高了遗传算法的搜索效率。

(4) 传统优化算法的搜索方法是确定性的方法,对算法的应用范围有一定的限制;而遗传算法的搜索方法是一种概率搜索法,它用概率的变化来引导遗传算法的搜索过程在搜索空间中移向更优化的解区域,并且其选择、交叉、变异等算子都是以概率的方式进行操作的,提高了搜索的灵活性。

(5) 遗传算法的容错能力很强。遗传算法的初始群体中的大部分个体与所求问题的最优解相离很远,但它能通过相关的遗传操作迅速减少与最优解相差极大的个体数。

## 2.2 基于遗传算法优化的常模盲均衡算法

与传统的自适应均衡技术相比,不需要训练序列的盲均衡技术能够有效节省通信中本不富裕的信道带宽,消除码间干扰,提高通信效率,从而越来越受人们关注。在盲均衡技术中,传统的常模盲均衡算法(constant modulus blind equalization algorithm,CMA)利用代价函数对均衡器权向量的梯度来确定均衡器权向量的迭代方程,但这种方法本质上是一种只考虑局部区域的梯度下降搜索法[9-10],缺乏全局搜索能力,同时构造的代价函数还须满足可导。

由前面的分析可知,遗传算法是一种群体搜索技术,它将一组问题的解用种群来表示,并对当前种群进行选择、交叉和变异等遗传操作来产生新一

代种群，从而逐步使种群进化到近似最优解，它不依赖于梯度信息，也不需要求解函数可微，是一种具有全局性和强鲁棒性的随机搜索方法。

现将遗传算法引入到常模盲均衡算法中，得到基于遗传算法优化的常模盲均衡算法(genetic optimization algorithm based CMA，GACMA)[11-14]，如图 2.1 所示。

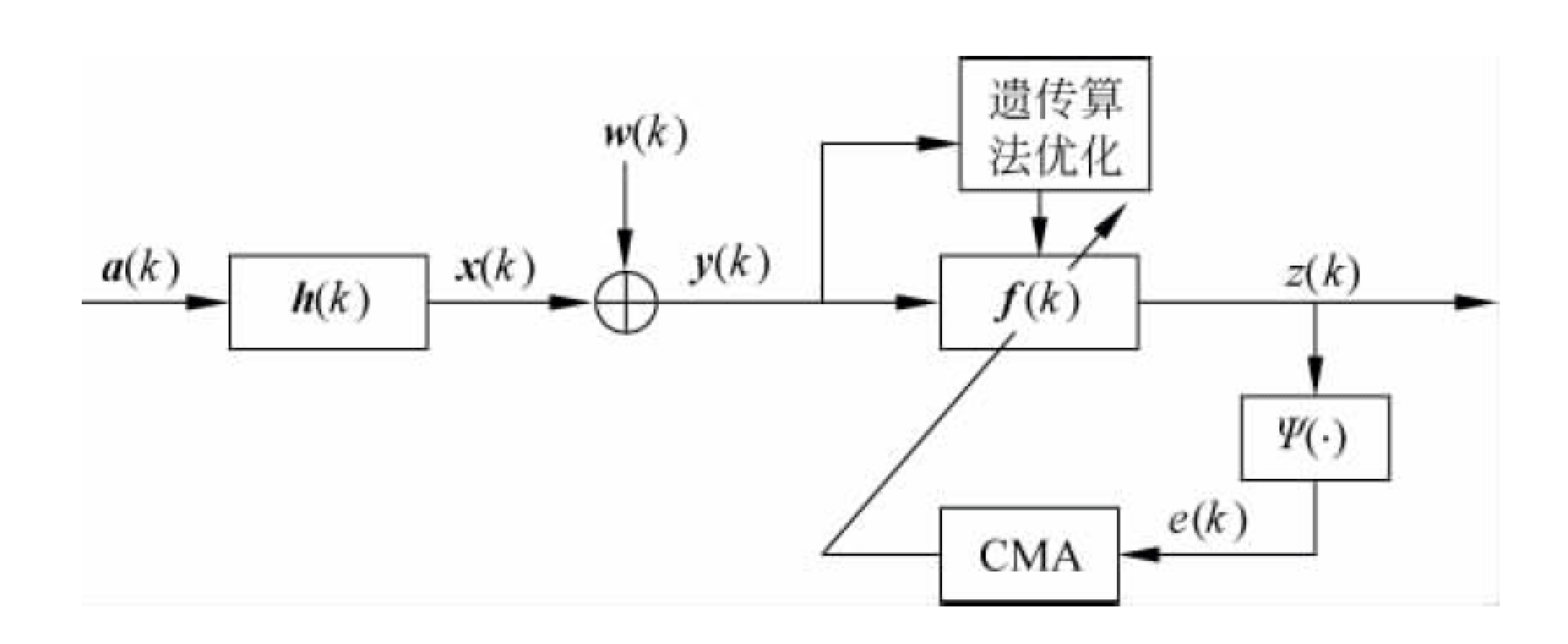



图 2.1 GACMA 原理

图 2.1 中，$\boldsymbol{a}(k)$是信源信号向量，也即信道输入信号向量；$\boldsymbol{h}(k)$表示信道冲激响应向量，反映了信道信息；$\boldsymbol{w}(k)$表示干扰噪声向量，一般为加性高斯白噪声(additional white gaussian noise，AWGN)；$\boldsymbol{y}(k)$是均衡器输入信号向量或信道输出含噪向量；$\boldsymbol{f}(k)$是均衡器权向量；$z(k)$是均衡器输出信号，最接近信源信号 $a(k)$，均衡后的 $z(k)$和 $a(k)$之间的误差非常小，$z(k)$经过判决后就能够准确地表示信源信号 $a(k)$；$\boldsymbol{\Psi}(\cdot)$是误差生成函数；$e(k)$是误差函数。

$$y(k) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k) + w(k) \tag{2.2.1}$$

$$z(k) = \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k) = \boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{2.2.2}$$

CMA 误差函数为

$$e(k) = |z(k)|^2 - R^2 \tag{2.2.3}$$

式中，$R^2$ 为 CMA 模值。

CMA 权向量迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - 2\mu e(k)z(k)\boldsymbol{y}^*(k) \tag{2.2.4}$$

在 GACMA 中，利用遗传算法寻找均衡器最优权向量的基本思想是：

把均衡器的权向量作为遗传算法的决策变量,把均衡器输入信号作为遗传算法的输入,由 CMA 代价函数定义遗传算法的适应度函数,利用遗传算法来求解均衡器代价函数全局最小值,从而得到均衡器权向量最优值。

均衡器的代价函数由均衡器误差的时间平均来表示,设接收信号序列的长度为 $N$,其代价函数定义为

$$J_{\mathrm{CMA}}(k) = \sum_{k=1}^{N}(|z(k)|^2 - R^2)^2/N \tag{2.2.5}$$

式中,$z(k)$是均衡器的输出;$R^2$ 是均衡器的模值。因此,遗传算法进化的每一代都将依次接收 $N$ 个输入信号,每一代中这 $N$ 个信号首先利用常模盲均衡算法来实现均衡,随后再进行遗传算法的进化操作,并将进化产生的新的种群作为下一代进化的初始种群,其具体的优化过程如下。

**步骤 1**:初始群体。遗传算法进化操作的对象是群体,因此进化操作前需要给其设置一些初始群体数据,这些初始群体数据用来表示起始搜索点。为了简单起见,初步确定遗传算法的解空间为[−1,1],并随机产生一定数目的个体,这些个体构成一个初始群体;其中每个个体对应均衡器的一个权向量,个体长度对应均衡器权向量的长度。本节个体的编码方法采用实数编码,编码值是[−1,1]内的一个随机数。设随机产生的一个初始种群为 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$,其中一个个体 $\boldsymbol{f}_i(0<i\leqslant M)$对应均衡器一个权向量。

**步骤 2**:确定适应度函数。盲均衡算法的目标是使代价函数迭代至全局极小值,得到均衡器的最优权值;而遗传算法求解的目标是得到适应度值最大的个体。为了解决这个矛盾,将均衡器代价函数的倒数作为适应度函数,则有

$$\mathrm{Fit}(\boldsymbol{f}_i) = \frac{1}{J(\boldsymbol{f}_i)}, \quad i = 1,2,\cdots,M \tag{2.2.6}$$

式中,$J(\boldsymbol{f}_i)$是均衡器的代价函数,$\boldsymbol{f}_i$ 是遗传算法产生的均衡器权向量个体。

**步骤 3**:设计遗传算子。遗传操作是利用个体的适应度函数进行的。它包括选择运算、交叉运算和变异运算。选择运算是利用群体中各个个体适应度值被选中的概率来决定其遗传到下一代的可能性。本节选择运算用

随机遍历抽样法代替常用的轮盘选择法,轮盘选择法使用的是单个选择指针,而随机遍历抽样法使用 $K$ 个相等距离的指针,这里的 $K$ 是要选择的权向量个体数,选择指针的距离是 $1/K$,第一个指针的位置由$[0,1/K]$区间的均匀随机数决定,这样 $K$ 个个体就由相隔一个指针距离的 $K$ 个指针选择,选择累积概率离指针位置近的权向量个体。其中,累积概率用权向量个体的选择概率求取。若有 $K$ 个权向量个体,第 $i$ 个权向量个体 $\boldsymbol{f}_i$ 的适应度值为 $\mathrm{Fit}(\boldsymbol{f}_i)$,其被选择的概率为

$$p(f_i) = \frac{\mathrm{Fit}(f_i)}{\sum_{i=1}^{K}\mathrm{Fit}(f_i)} \tag{2.2.7}$$

则第 $n$ 个权向量个体的累积概率为

$$q_n = \sum_{i=1}^{n} p(f_i), \quad (1 \leqslant n \leqslant K) \tag{2.2.8}$$

交叉运算在遗传操作中起核心作用,是产生新的权向量个体的主要方法,考虑到权向量个体采用的是实数编码,为保证交叉后产生新的参数值,并开辟新的搜索空间,于是交叉操作采用两点交叉和线性组合方法,在相互配对的两个权向量个体编码串中随机设置两个交叉点,对交叉位的编码值进行线性组合生成新的权向量个体。设进行交叉的两个父代个体分别为 $f_i$ 和 $f_{i+1}$,线性组合后得到的子代个体 $f'_i$和 $f'_{i+1}$为

$$f'_i = f_i + \alpha(f_{i+1} - f_i) \tag{2.2.9a}$$

$$f'_{i+1} = f_{i+1} + \alpha(f_i - f_{i+1}) \tag{2.2.9b}$$

式中,$\alpha$ 是一个比例因子,可由$[0,1]$上均匀分布的随机数产生。

变异运算在遗传操作中属于辅助性的搜索操作,实现从局部的角度出发使个体更加逼近最优解。此处,对于实数编码的权向量个体采用实值变异,设 $f_i(m)$是变异前的第 $i$ 个权向量个体的第 $m$ 个抽头值,$f'_i(m)$是变异后的第 $i$ 个权向量个体的第 $m$ 个抽头值,则有

$$f'_i(m) = f_i(m) \pm 0.5L\Delta \tag{2.2.10}$$

式中,$\Delta = \sum_{t=0}^{m-1}\frac{B(t)}{2^t}$,$B(t)$以概率 $1/m$ 取值 1,以概率 $1-1/m$ 取值 0;$L$ 为

权值的取值范围。

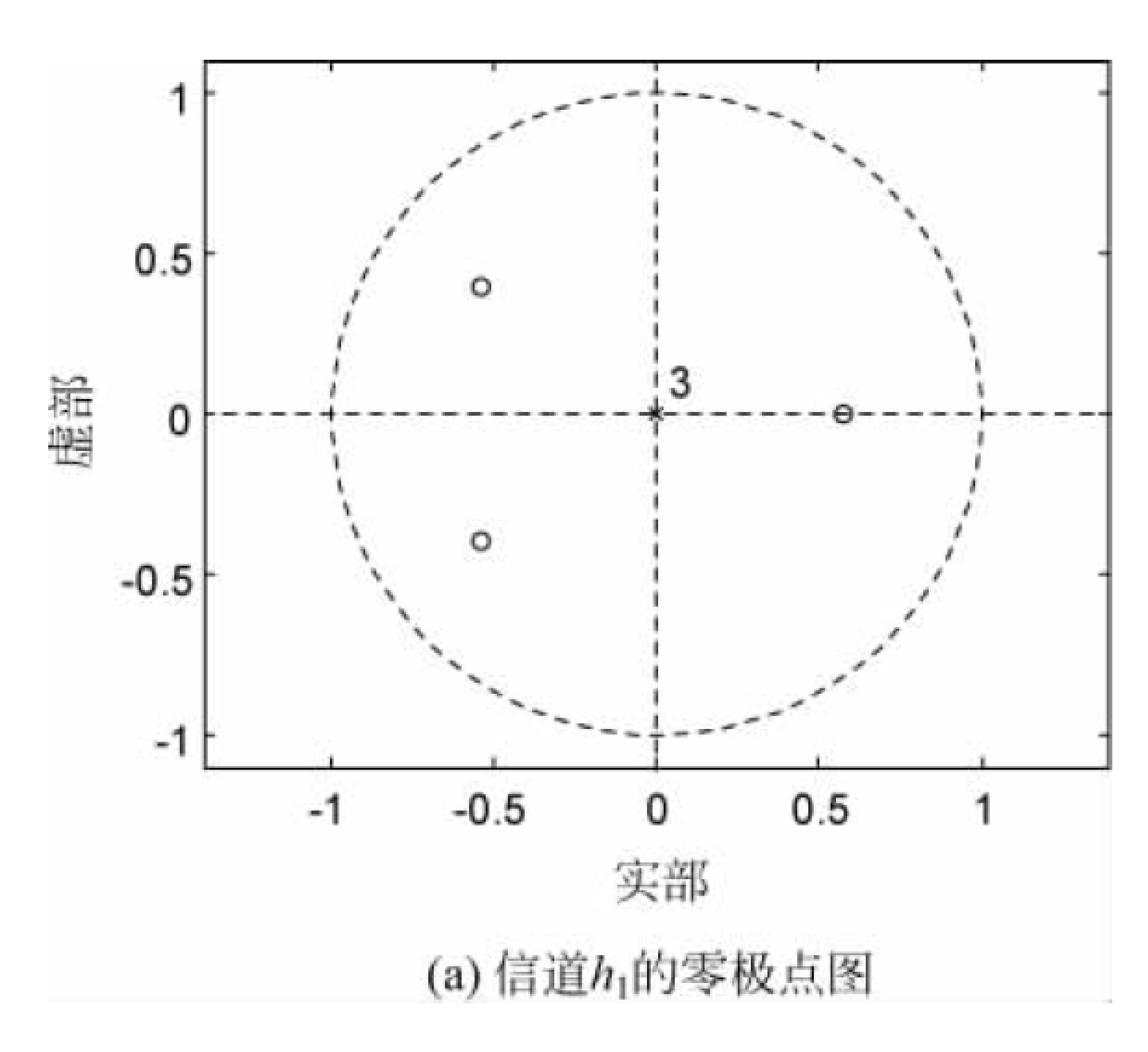
**步骤4**：终止与否的判断。遗传算法的终止用最大进化代数来判断，当进化代数不大于最大进化代数，则返回步骤2，反复进行步骤2～步骤4操作，直至进化代数大于最大进化代数，否则进行步骤5操作。

**步骤5**：选择最佳的权向量个体。将每代中适应度值最大的权向量个体选择出来，考虑到算法在抽取最佳个体时的实时性和盲均衡算法要满足迫零条件，抽取最佳权向量个体时将本代的最佳权向量个体作为下一代的最佳权向量个体输出。

以上是将遗传算法引入到常模盲均衡算法的思想、方法与过程，通过遗传算法来优化均衡器权向量。这种引入遗传算法的常模盲均衡算法，称为基于遗传算法优化的常模盲均衡算法（GACMA）。

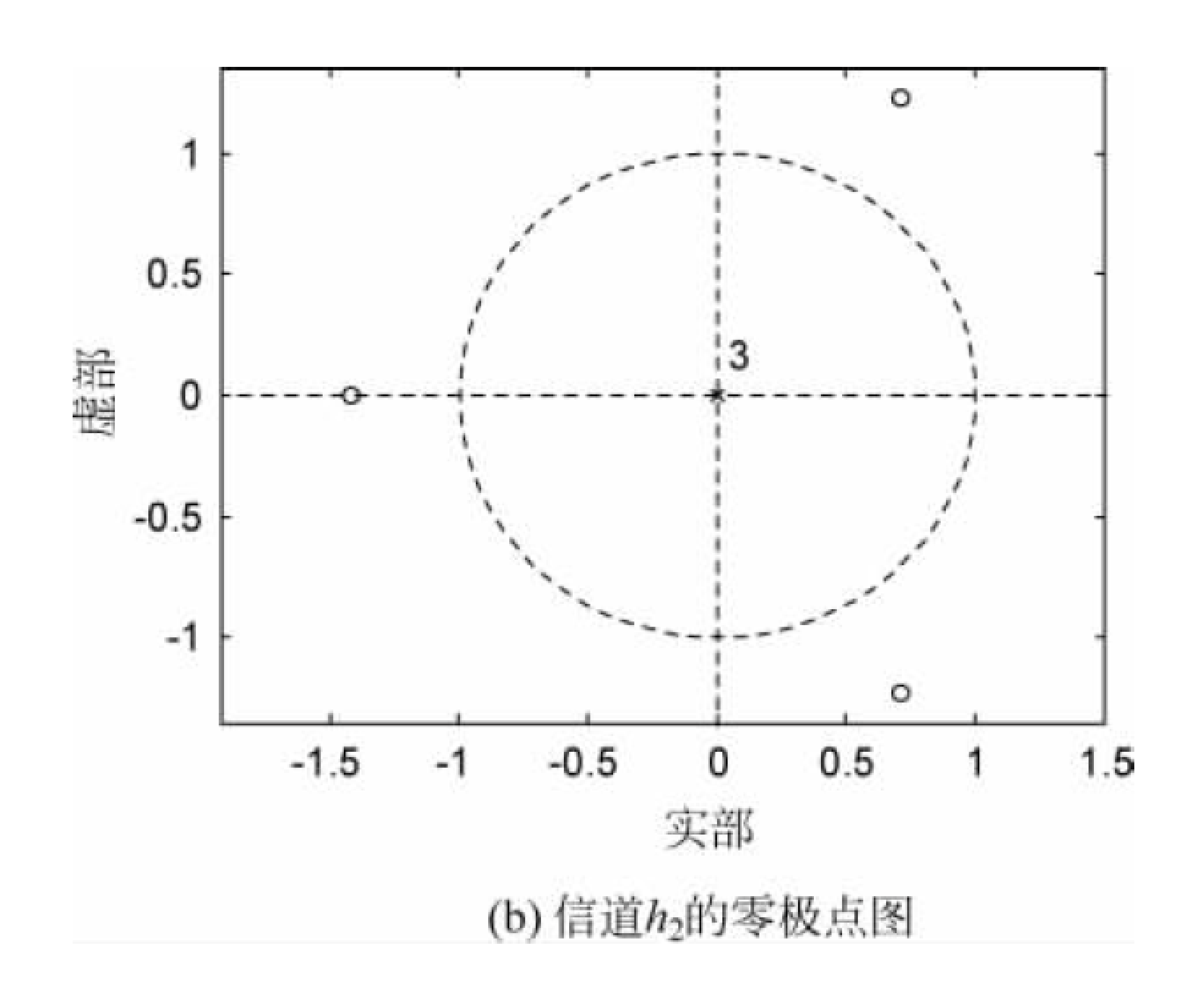
为了检验GACMA的有效性，以CMA为比较对象，进行仿真实验。

**【实验2.1】** 该实验采用一条突变信道，即由两条信道[10]进行组合的方法来构造。发射信号先通过信道$h_1=[0.3122\quad -0.1040\quad 0.8908\quad 0.3134]$，在第$c_1$个采样点的时刻，信道发生变化，变为信道$h_2=[0.35\quad 0\quad 0\quad 1]$，两条信道各有不同的零极点特性，在不同的时间段信号通过不同的信道。发射机一共发射8000个信号采样点，$c_1=3900$。两条信道的零极点如图2.2所示。



(a) 信道$h_1$的零极点图

图2.2　两信道零极点图



(b) 信道$h_2$的零极点图

图 2.2 （续）

发射信号为 8PSK 时，信噪比为 20dB，均衡器权长是 16；步长 $\mu_{CMA}=0.01$，$\mu_{GACMA}=0.008$；CMA 中第 4 个抽头系数设置为 1，其余为 0；遗传算法的交叉概率为 0.7，变异概率为 1/16，进化代数为 100。500 次蒙特卡罗仿真结果，如图 2.3 所示。

图 2.2 表明，随着对发射信号不同时刻的采样，信道的零点正在由单位圆内向外散开。所以，由这两条信道模拟一条时变信道，可以较好地体现信道的时变特性。

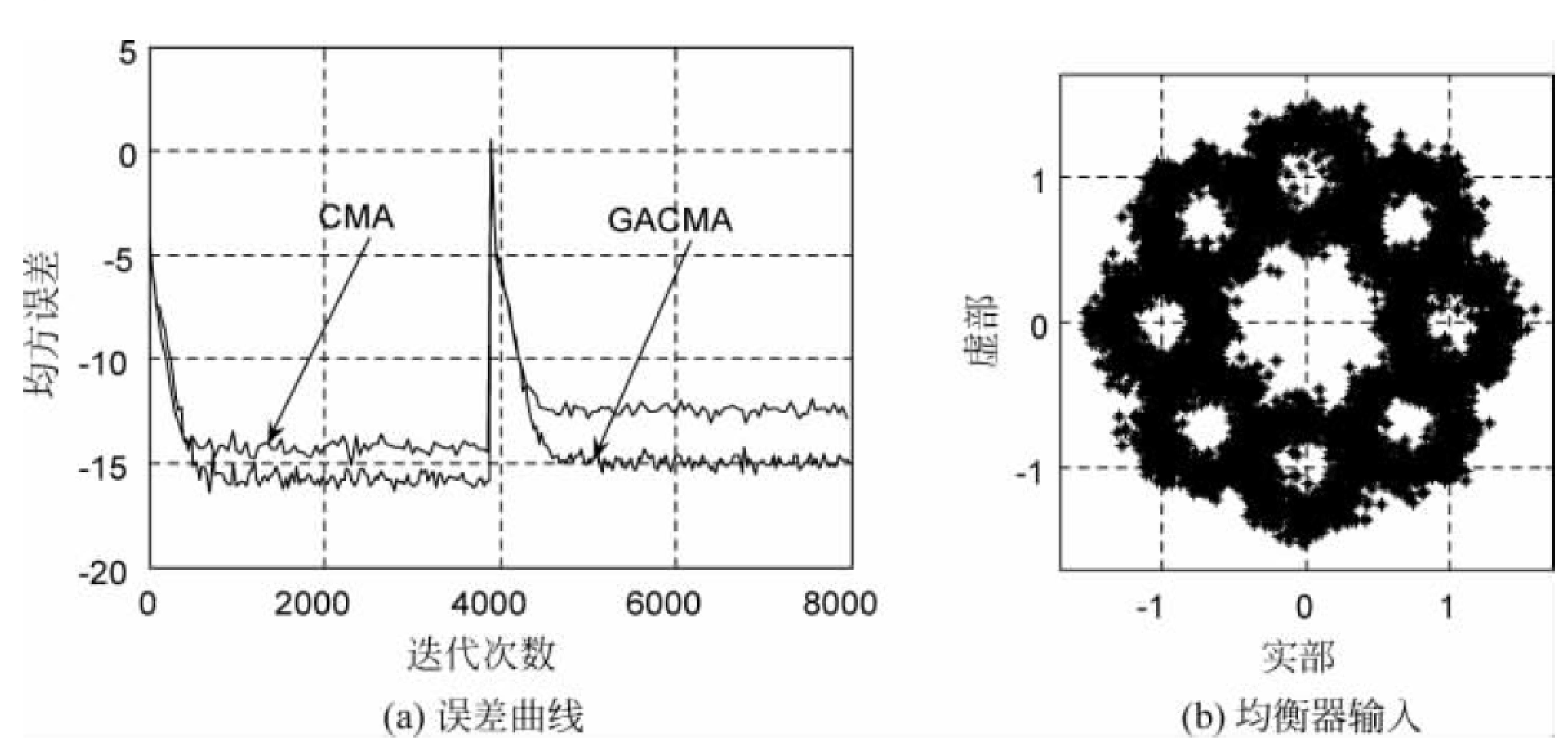
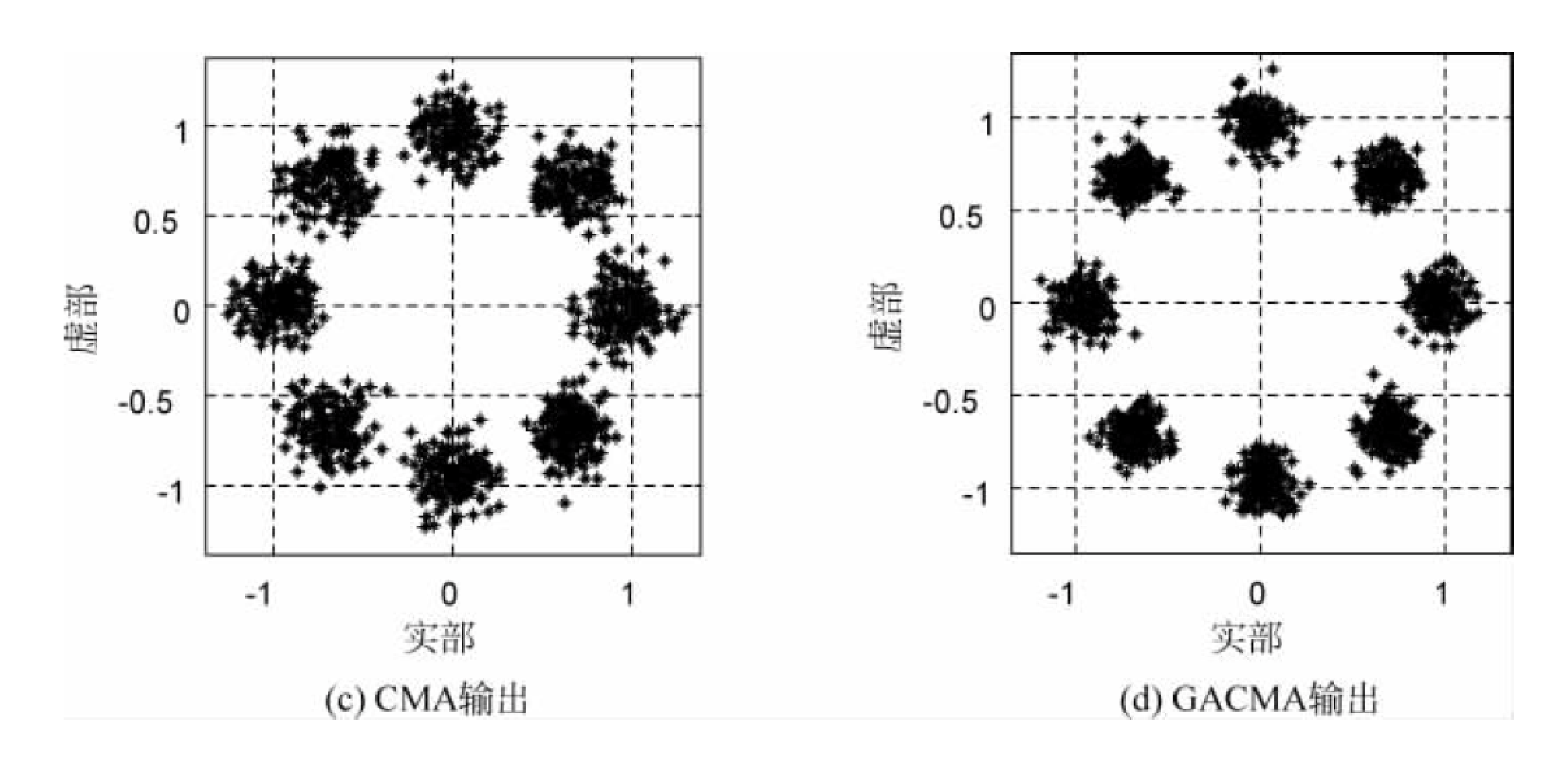



图 2.3 仿真结果



(c) CMA输出

(d) GACMA输出

图 2.3 （续）

图 2.3 表明，当时变信道发生变化时，与 CMA 相比，GACMA 有较强的重新启动能力，能快速跟踪信道的时变特性。而且，GACMA 输出星座图更加紧密集中，眼图张开更加清晰、更具有实用价值。

## 2.3 基于遗传算法优化的正交小波常模盲均衡算法

传统的常数模盲均衡算法的收敛速度较慢、稳态误差较大；而盲均衡算法的性能与均衡器输入信号的自相关性有着一定的关系，自相关性小、收敛速度快。研究表明，利用正交小波基函数对均衡器输入信号进行正交小波变换并作能量归一化处理，信号与噪声的相关性会得到一定程度的降低，能有效加快收敛速度。然而，传统的正交小波盲均衡算法依然是按随机梯度搜索来寻找最优权向量，仍容易收敛至局部极小值，甚至发散。本节将充分利用遗传算法的全局随机搜索特点，快速寻找均衡器权向量的全局最优解，避免权向量陷入局部最优解。

### 2.3.1 基于正交小波变换的常模盲均衡算法

文献[15～17]表明，基于正交小波变换的常模盲均衡算法(orthogonal

wavelet transform based constant modulus blind equalization algorithm, WTCMA),具有快速收敛性能,其原理如图 2.4 所示。

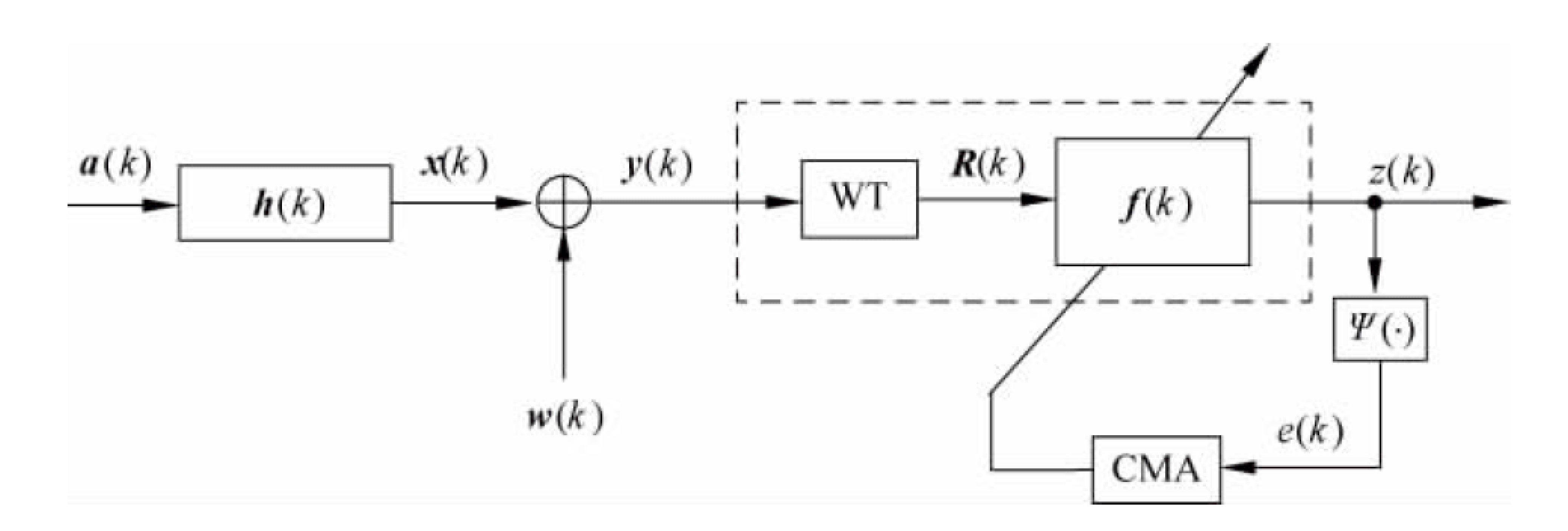



图 2.4 WTCMA 原理

图中,$\boldsymbol{a}(k)$为发射信号,$\boldsymbol{h}(k)$为信道脉冲响应向量,$\boldsymbol{w}(k)$为高斯白噪声向量,$\boldsymbol{y}(k)$为均衡器的接收信号向量;WT 表示正交小波变换;$\boldsymbol{\Psi}(\cdot)$是误差生成函数,$e(k)$是误差函数;$\boldsymbol{f}(k)$是均衡器权向量;$z(k)$是均衡器输出信号。

由分析小波理论可知,当均衡器权向量 $\boldsymbol{f}(k)$为有限冲击响应时,$\boldsymbol{f}(k)$可用一组正交小波基函数来表示:

$$\boldsymbol{f}(k) = \sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{jn} \cdot \varphi_{jn}(k) + \sum_{n=0}^{k_j} v_{jn} \cdot \phi_{jn}(k) \tag{2.3.1}$$

式中,$n=0,1,\cdots,N-1$;$J$ 表示小波分解的最大尺度;$k_j=N/2^j-1(j=1,2,\cdots,J)$表示尺度 $j$ 下小波函数的最大平移;$d_{jn}$和 $v_{jn}$ 表示均衡器的权系数;$\varphi_{jn}(k)$和 $\phi_{jn}(k)$分别表示小波函数和尺度函数。根据信号传输理论,均衡器输出为

$$z(k) = \sum_{n=0}^{N-1} f_n(k) \cdot y(k-n) = \sum_{j=1}^{J}\sum_{n=0}^{k_j} d_{jn} \cdot u_{jn}(k) + \sum_{n=0}^{k_j} v_{jn} \cdot s_{jn}(k) \tag{2.3.2}$$

式中,$u_{j,n}(k)$为尺度为 $j$ 平移为 $n$ 的小波变换系数;$s_{J,n}(k)$为尺度为 $J$ 平移为 $n$ 的尺度变换系数。式(2.3.2)表明,$\boldsymbol{f}(k)$用小波基函数表示的实质是对均衡器的输入信号进行正交小波变换,改变了均衡器的结构。

将均衡器的正交小波变换系数记为

$$\boldsymbol{R}(k)=[u_{1,0}(k),u_{1,1}(k),\cdots,u_{J,K_J}(k),s_{J,0}(k),s_{J,1}(k),\cdots,s_{J,K_J}(k)]^{\mathrm{T}} \tag{2.3.3}$$

均衡器未知权向量为

$$\boldsymbol{f}(k)=[d_{1,0}(k),d_{1,1}(k),\cdots,d_{J,K_J}(k),v_{J,0}(k),v_{J,1}(k),\cdots,v_{J,K_J}(k)]^{\mathrm{T}} \tag{2.3.4}$$

设 $\boldsymbol{V}$ 为正交小波变换矩阵，则经过正交小波变换后均衡器的输入为

$$\boldsymbol{R}(k)=\boldsymbol{V}\boldsymbol{y}(k) \tag{2.3.5}$$

均衡器的输出

$$z(k)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{R}(k) \tag{2.3.6}$$

均衡器的误差

$$e(k)=R^2-|z(k)|^2 \tag{2.3.7}$$

均衡器权向量的迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu\hat{\boldsymbol{R}}^{-1}(k)e(k)\boldsymbol{R}^*(k) \tag{2.3.8}$$

式中，$\hat{\boldsymbol{R}}(k)=\mathrm{diag}[\sigma_{j,0}^2(k),\sigma_{j,1}^2(k),\cdots,\sigma_{J,k_J}^2(k),\sigma_{J+1,0}^2(k),\cdots,\sigma_{J+1,k_J}^2(k)]$；$\mu$ 为迭代步长；$\sigma_{j,n}^2(k)$、$\sigma_{J+1,n}^2(k)$分别表示对小波变换系数 $u_{j,n}(k)$、尺度变换系数 $s_{J,n}(k)$的平均功率估计，其递推公式为

$$\hat{\sigma}_{j,n}^2(k+1)=\beta\hat{\sigma}_{j,n}^2(k)+(1-\beta)|u_{j,n}(k)|^2 \tag{2.3.9}$$

$$\hat{\sigma}_{J+1,n}^2(k+1)=\beta\hat{\sigma}_{J+1,n}^2(k)+(1-\beta)|s_{J,n}(k)|^2 \tag{2.3.10}$$

式中，$\beta$ 为平滑因子，且 $0<\beta<1$，一般取 $\beta$ 值比较接近于 1。称式(2.3.1)～式(2.3.10)构成的算法为基于正交小波变换的常模盲均衡算法(WTCMA)。WTCMA 是基于 CMA 的思想对均衡器权向量进行更新的，即先构造一个代价函数，利用局部区域的梯度下降搜索法确定均衡器权值的迭代方程，全局搜索能力较差，易陷入局部收敛。

### 2.3.2 基于遗传算法优化的正交小波常模盲均衡算法

#### 1. 算法描述

为了克服 WTCMA 的缺陷，本节将遗传算法引入到 WTCMA 中。遗

传算法的输入信号由经过正交小波变换后均衡器输入信号提供，决策变量对应于均衡器的权向量，它通过随机方法产生，适应度函数对应于 CMA 代价函数的倒数，利用遗传算法求解均衡器的代价函数，寻找最优权向量。将每一代中适应度值最大的权向量个体（称为最佳个体）选择出来，考虑到算法在抽取最佳个体时的实时性和盲均衡算法要满足迫零条件，抽取最佳权向量个体时将本代的最佳权向量个体作为下一代的最佳权向量个体输出。这样，将遗传算法引入到正交小波变换的常模盲均衡算法，得到基于遗传算法优化的正交小波常模盲均衡算法（genetic optimization algorithm based WTCMA，GA-WTCMA）[11,18,19]。其原理如图 2.5 所示。优化过程类似于 2.2 节，这里不再赘述。

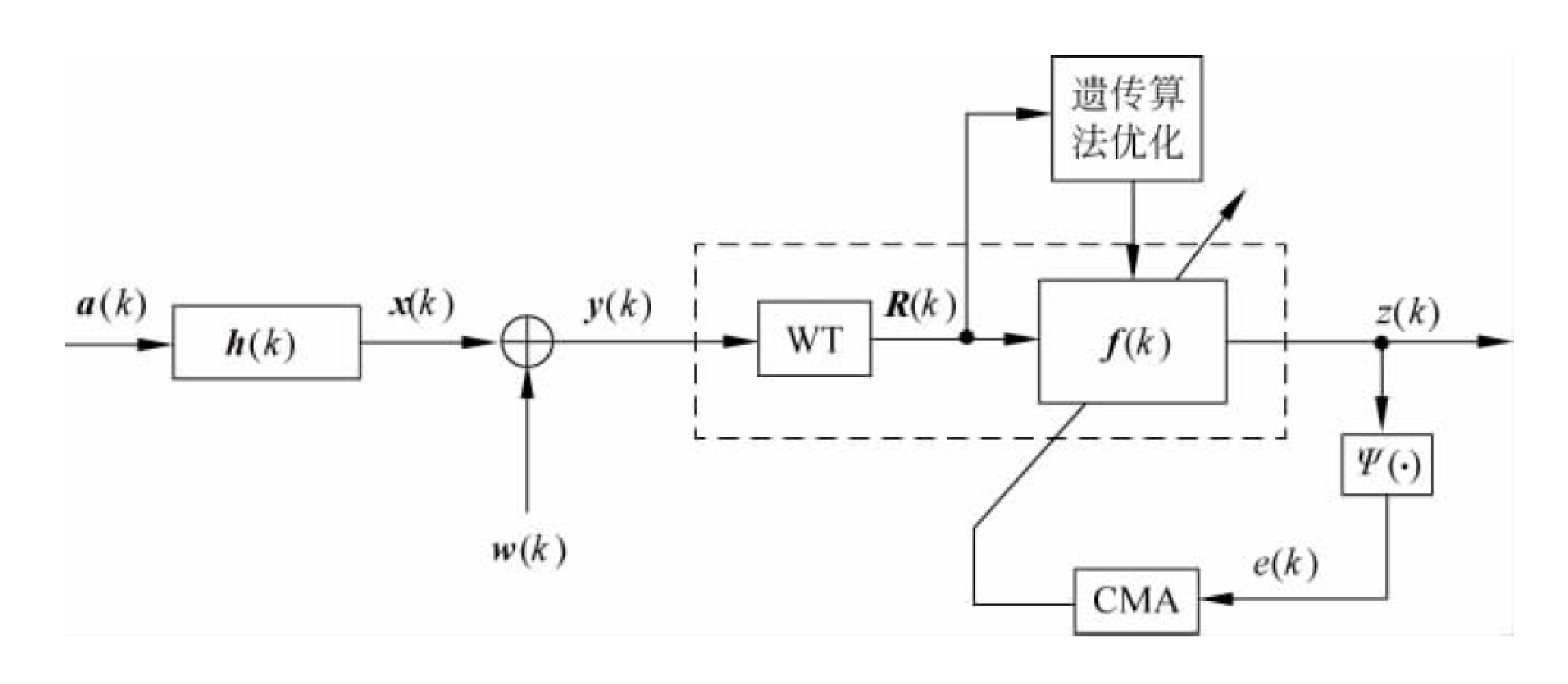



图 2.5　GA-WTCMA 原理

2. 算法仿真

为了检验 GA-WTCMA 的有效性，以 WTCMA 为比较对象，进行仿真实验。

**【实验 2.2】** 最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$[10]，发射信号为 16QAM 信号；信噪比为 25dB；均衡器权长为 32；功率初始值设置为 4，种群规模为 100，交叉概率为 0.7，变异概率为 1/32，最大进化代数为 160。WTCMA 采用第 5 个抽头初始化为 1，其他参数设置如表 2.1 所示。500 次蒙特卡罗仿真结果如图 2.6 所示。

表 2.1　仿真参数设置

| 算　　法 | 仿真步长 | 小波 | 分解层数 | $\beta$取值 |
|---|---|---|---|---|
| WTCMA | 0.000 25 | DB2 | 3 | 0.9999 |
| GA-WTCMA | 0.0002 | DB2 | 3 | 0.9999 |

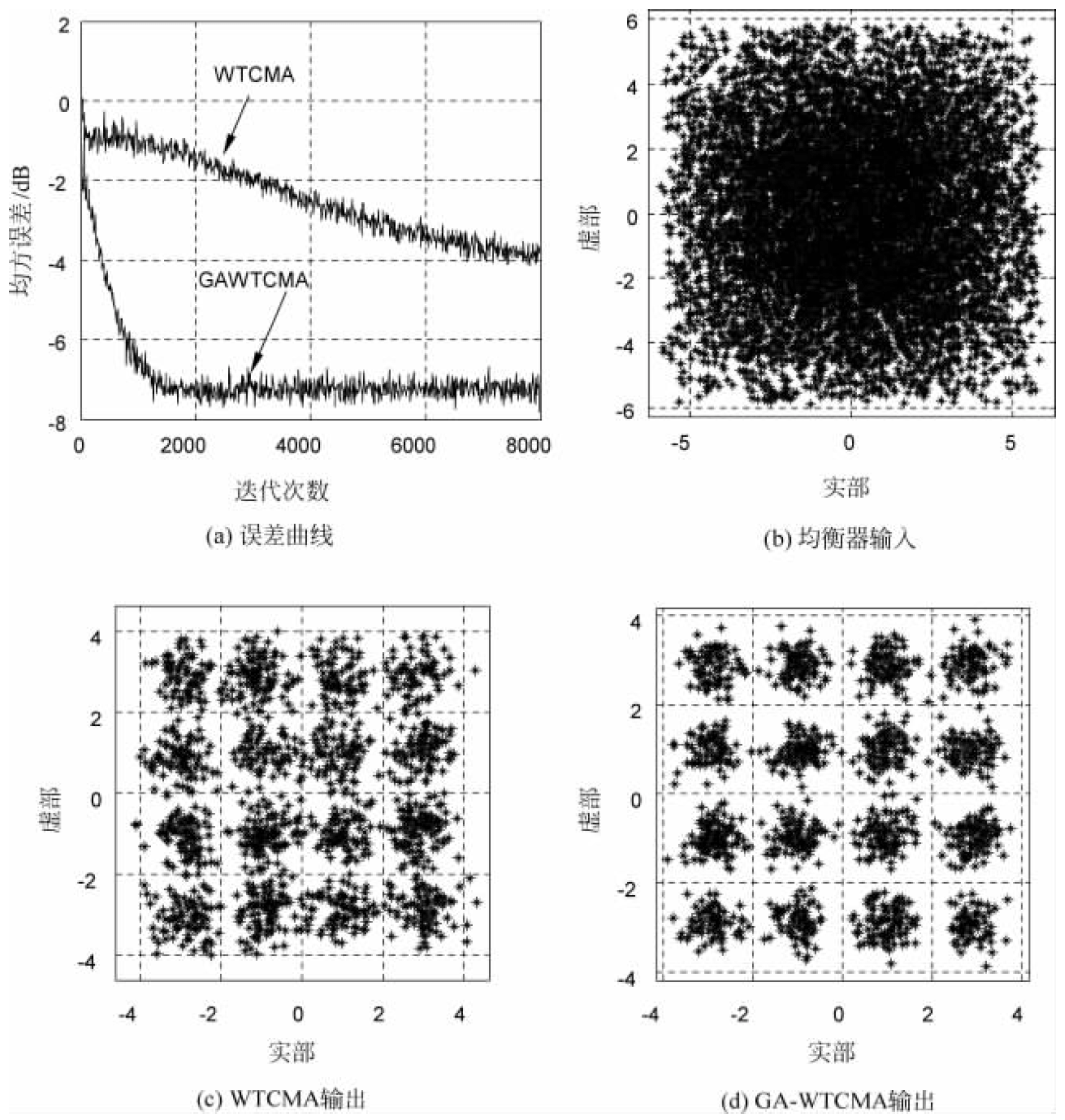



图 2.6　仿真结果

图 2.6 表明，在收敛速度上，GA-WTCMA 比 WTCMA 快了近 5000 步；在稳态误差上，GA-WTCMA 比 WTCMA 减小了 3dB；与 WTCMA 相比，GA-WTCMA 的输出星座图明显更加清晰、紧凑。

## 2.4 基于自适应遗传算法优化的正交小波常模盲均衡算法

### 2.4.1 自适应遗传算法

自适应遗传算法[20-22]（adaptive genetic algorithm，AGA）中交叉概率 $p_c$ 和变异概率 $p_m$ 是自适应改变的，其改变方法是借助个体适应度值来改变。当群体的优化解趋于所求问题的局部最优解时，就相应地增大 $p_c$ 和 $p_m$；当群体的优化解在解空间中趋于发散时，就降低 $p_c$ 和 $p_m$。同时，当个体的适应度值高于群体的平均适应度值时，则 $p_c$ 和 $p_m$ 取值较低，使得该个体对应的所求解能进入下一代；而个体的适应度值低于平均适应度值时，则 $p_c$ 和 $p_m$ 取值较高，使得该解被淘汰掉。因此，自适应的 $p_c$ 和 $p_m$ 能够为所求问题的某个解提供最佳 $p_c$ 和 $p_m$。自适应遗传算法进化初期在大范围内对种群进行全局搜索以避免早熟收敛；进化后期，搜索的解逼近最优解，种群的搜索应在局部范围内，以提高算法的精度。自适应遗传算法不仅能保持群体多样性，而且能保证遗传算法的收敛性，从而使遗传算法的搜索优化能力得到有效提高。

在自适应遗传算法中，交叉概率和变异概率的自适应调整公式为

$$p_c = \begin{cases} \dfrac{k_1(\mathrm{Fit}_{\max} - \mathrm{Fit}')}{\mathrm{Fit}_{\max} - \mathrm{Fit}_{\mathrm{avg}}} & \mathrm{Fit} \geqslant \mathrm{Fit}_{\mathrm{avg}} \\ k_2 & \text{其他} \end{cases} \tag{2.4.1}$$

$$p_m = \begin{cases} \dfrac{k_3(\mathrm{Fit}_{\max} - \mathrm{Fit})}{\mathrm{Fit}_{\max} - \mathrm{Fit}_{\mathrm{avg}}} & \mathrm{Fit} \geqslant \mathrm{Fit}_{\mathrm{avg}} \\ k_4 & \text{其他} \end{cases} \tag{2.4.2}$$

式中，$\mathrm{Fit}_{\max}$ 为当代种群中适应度最大值；$\mathrm{Fit}_{\mathrm{avg}}$ 为每代群体的平均适应度值；$\mathrm{Fit}'$ 为每一代要交叉的两个个体中的适应度较大值，Fit 为每代要变异的个体适应度值；$k_1$、$k_2$、$k_3$、$k_4$ 取(0,1)区间的一个值，只要调整 $k_1$、$k_2$、$k_3$、$k_4$ 就能实现交叉概率与变异概率的自适应调整，如图 2.7 所示。

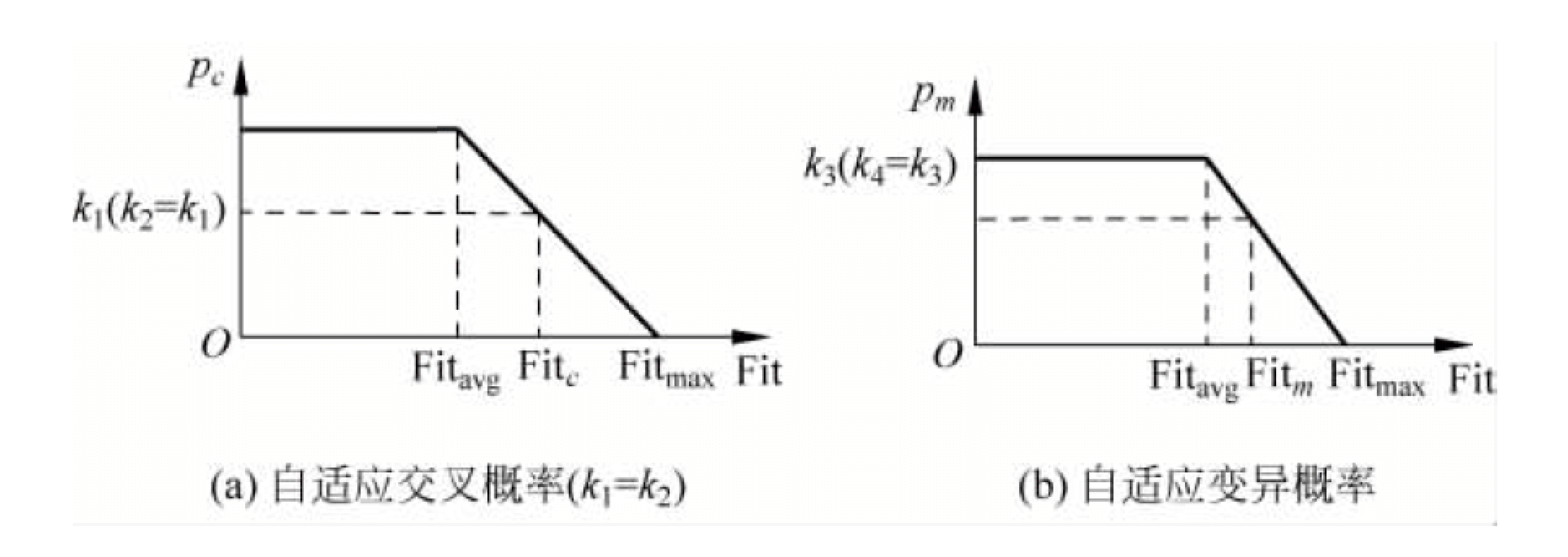



图 2.7　自适应交叉概率与变异概率

由上面的式子可知，当个体的适应度值低于当代群体的平均适应度值时，表明该个体的性能不好，于是对该个体采用较大的交叉概率和变异概率；如果个体的适应度值高于当代群体的平均适应度值，说明该个体性能优良，对它就根据其适应度值取相应的交叉概率和变异概率。可见，个体的适应度值越趋近于群体的适应度最大值，该个体的交叉概率、变异概率取值就越小；当个体的适应度值等于群体的适应度最大值时，交叉概率、变异概率的取值为零。这种借助于适应度值来调整交叉变异概率的方法比较适合于遗传算法的进化后期，这是因为进化后期群体中每个个体的性能基本上都比较好，这时不适合对个体进行较大的调整，以免使个体的优良性能结构遭到破坏；但在群体进化初期，这种自适应方法调整进化过程的效果不是很明显，群体中的较优良个体几乎不发生变化，且此时优良个体不一定就是所求问题的全局最优解，从而增加进化过程陷入局部最优解的可能性。

针对上面的问题，对式(2.4.1)和式(2.4.2)做进一步改进[22-23]，使进化过程中群体适应度最大值对应的个体交叉概率 $p_c$ 和变异概率 $p_m$ 的值不为零，并分别提高为某一 $p_c$ 和 $p_m$ 值，从而使群体中优良个体的交叉概率和变异概率得到相应提高，使这些优良个体不会停滞不前。本节根据适应度的相似度来自适应调整群体的交叉概率和变异概率，以群体的适应度最大值、最小值和平均值作为种群相近程度的衡量参量。交叉概率与变异概率分别为

$$p_c = \begin{cases} p_{c1} - \dfrac{p_{c1} - p_{c2}}{1 - \dfrac{\text{Fit}_{\min}}{\text{Fit}_{\max}}} & \dfrac{\text{Fit}_{\text{avg}}}{\text{Fit}_{\max}} > a, \dfrac{\text{Fit}_{\min}}{\text{Fit}_{\max}} > b \\ p_{c1} & \text{其他} \end{cases} \tag{2.4.3}$$

$$p_m = \begin{cases} p_{m1} - \dfrac{p_{m1} - p_{m2}}{1 - \dfrac{\text{Fit}_{\min}}{\text{Fit}_{\max}}} & \dfrac{\text{Fit}_{\text{avg}}}{\text{Fit}_{\max}} > a, \dfrac{\text{Fit}_{\min}}{\text{Fit}_{\max}} > b \\ p_{m1} & \text{其他} \end{cases} \tag{2.4.4}$$

式中，$p_{c1}$、$p_{m1}$为种群初始交叉、变异概率；$p_{c2}$、$p_{m2}$为提高后的种群交叉、变异概率；$\text{Fit}_{\max}$为群体的适应度最大值；$\text{Fit}_{\text{avg}}$为群体的适应度平均值；$\text{Fit}_{\min}$为群体的适应度最小值；$0.5<a<1$，$0<b<0.5$。

### 2.4.2 正交小波超指数迭代常模盲均衡算法

将超指数迭代算法引入到基于正交小波变换的常模盲均衡算法中，得到基于正交小波变换的超指数迭代常模盲均衡算法（super-exponential iterative blind equalization algorithm based on orthogonal wavelet transform，WTSEI）[24,25]，其原理如图2.8所示。

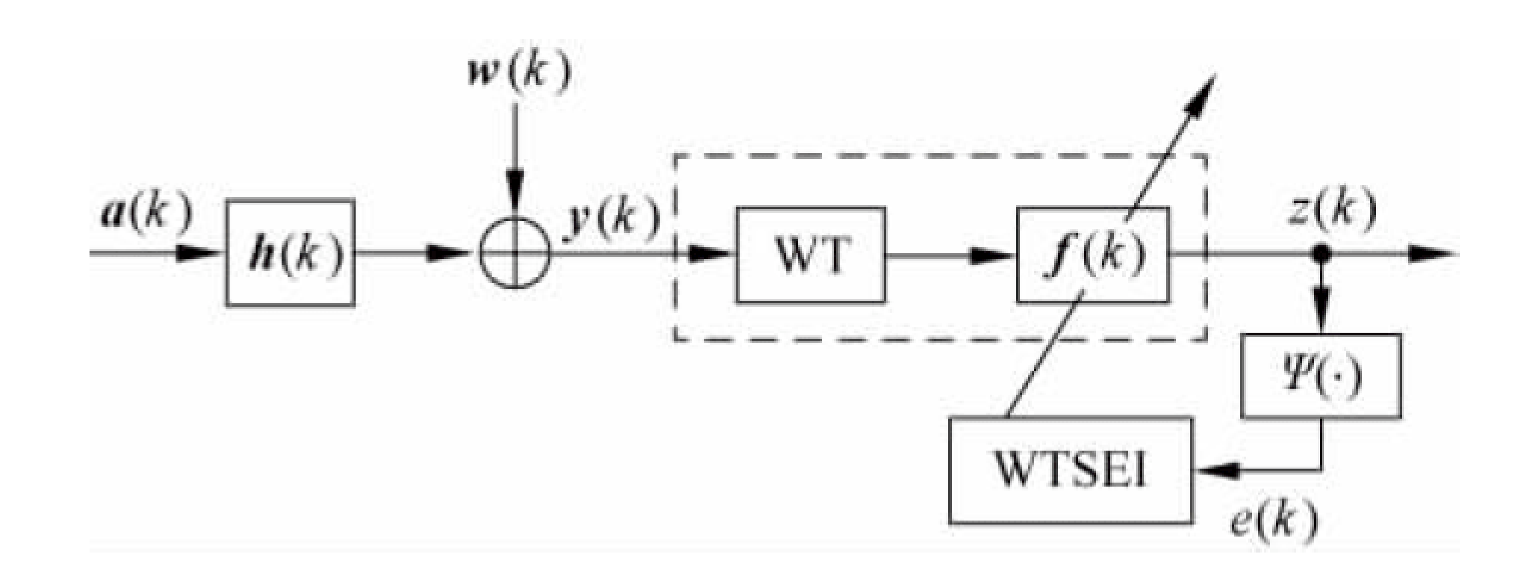



图2.8 WTSEI原理

图2.8中，$\boldsymbol{a}(k)$是独立同分布的发射信号；$\boldsymbol{h}(k)$为信道脉冲响应向量；$\boldsymbol{w}(k)$为高斯白噪声序列；$\boldsymbol{y}(k)$为均衡器的输入序列；$\boldsymbol{f}(k)$为均衡器权向量；$\Psi(\cdot)$是误差生成函数；$e(k)$为误差函数；$z(k)$是均衡器的输出序列。

设$\boldsymbol{V}$为正交小波变换矩阵，则经过小波变换后均衡器的输入为

$$\boldsymbol{R}(k) = \boldsymbol{y}(k)\boldsymbol{V} \tag{2.4.5}$$

均衡器的输出

$$z(k)=\boldsymbol{f}^{\mathrm{H}}(k)\boldsymbol{R}(k) \tag{2.4.6}$$

SEI 的白化矩阵 $\boldsymbol{Q}$ 的迭代公式为

$$\boldsymbol{Q}(k+1)=\frac{1}{1-\mu_m}\left[\boldsymbol{Q}(k)-\frac{\mu_m\boldsymbol{Q}(k)\boldsymbol{R}^*(k)\boldsymbol{R}^{\mathrm{T}}(k)\boldsymbol{Q}(k)}{1-\mu_m+\mu_m\boldsymbol{R}^{\mathrm{T}}(k)\boldsymbol{Q}(k)\boldsymbol{R}^*(k)}\right] \tag{2.4.7}$$

式中，$\mu_m$ 表示 $\boldsymbol{Q}$ 矩阵计算的迭代步长。权向量的迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu\hat{\boldsymbol{R}}^{-1}(k)\boldsymbol{Q}(k)e(k)\boldsymbol{R}(k)z(k) \tag{2.4.8}$$

称式(2.4.5)～式(2.4.8)构成了 WTSEI 算法。

### 2.4.3 基于自适应遗传算法优化的正交小波超指数迭代常模盲均衡算法

#### 1. 算法原理

在自适应遗传算法中，交叉概率 $p_c$ 和变异概率 $p_m$ 采用自适应选择的策略，使适应度高的个体结构不被破坏，保持了种群的多样性；使搜索过程不会停滞不前，有利于均衡器最佳权向量的搜索。于是本节将交叉概率和变异概率的自适应选择方法引入到正交小波超指数迭代常模盲均衡算法中，改变了基本遗传算法中采用反复实验来确定交叉概率和变异概率的方法，在保持群体多样性的同时，保证了遗传算法的收敛性，更好地优化了盲均衡算法的性能。这样得到基于自适应遗传算法优化的正交小波超指数迭代常模盲均衡算法[11] (adaptive genetic algorithm based WTSEI，AGA-WTSEI)原理，如图 2.9 所示；其自适应遗传算法优化流程如图 2.10 所示。

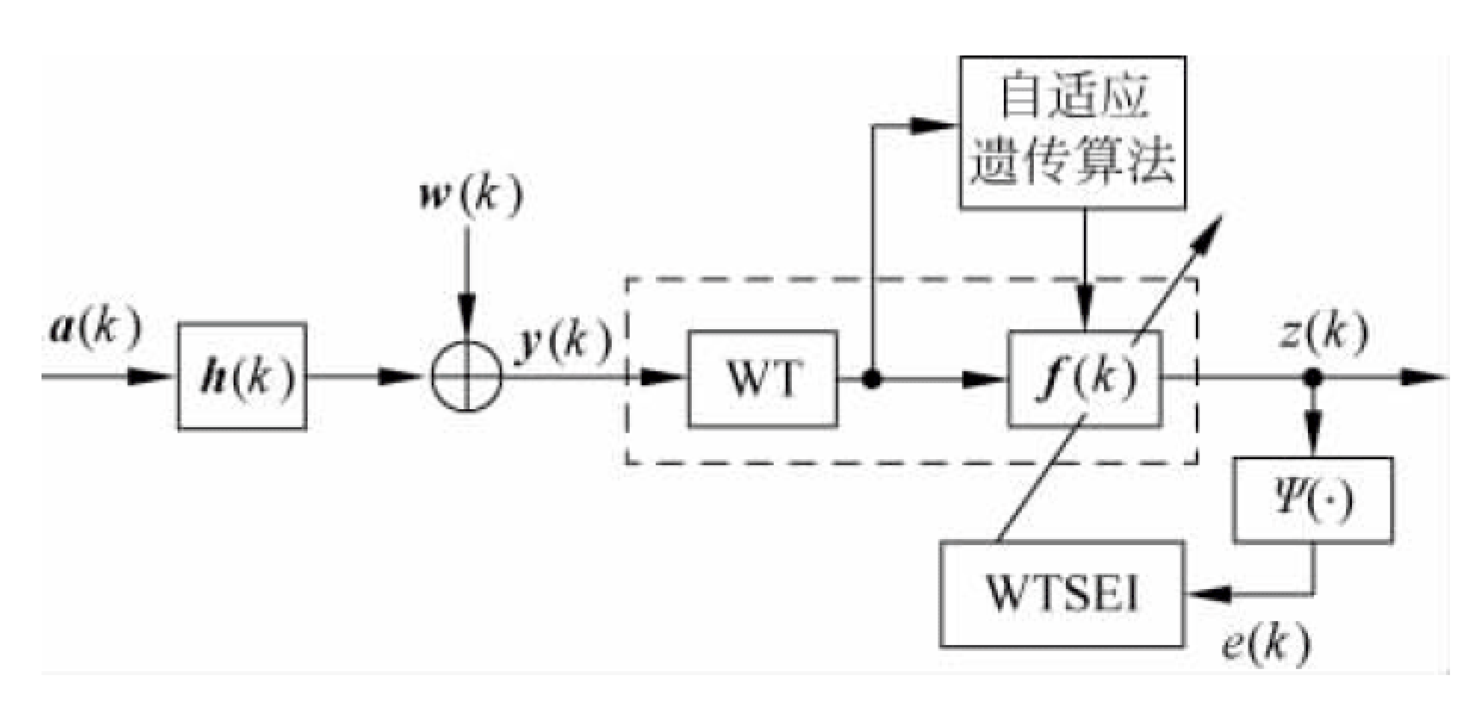



图 2.9　AGA-WTSEI 算法原理

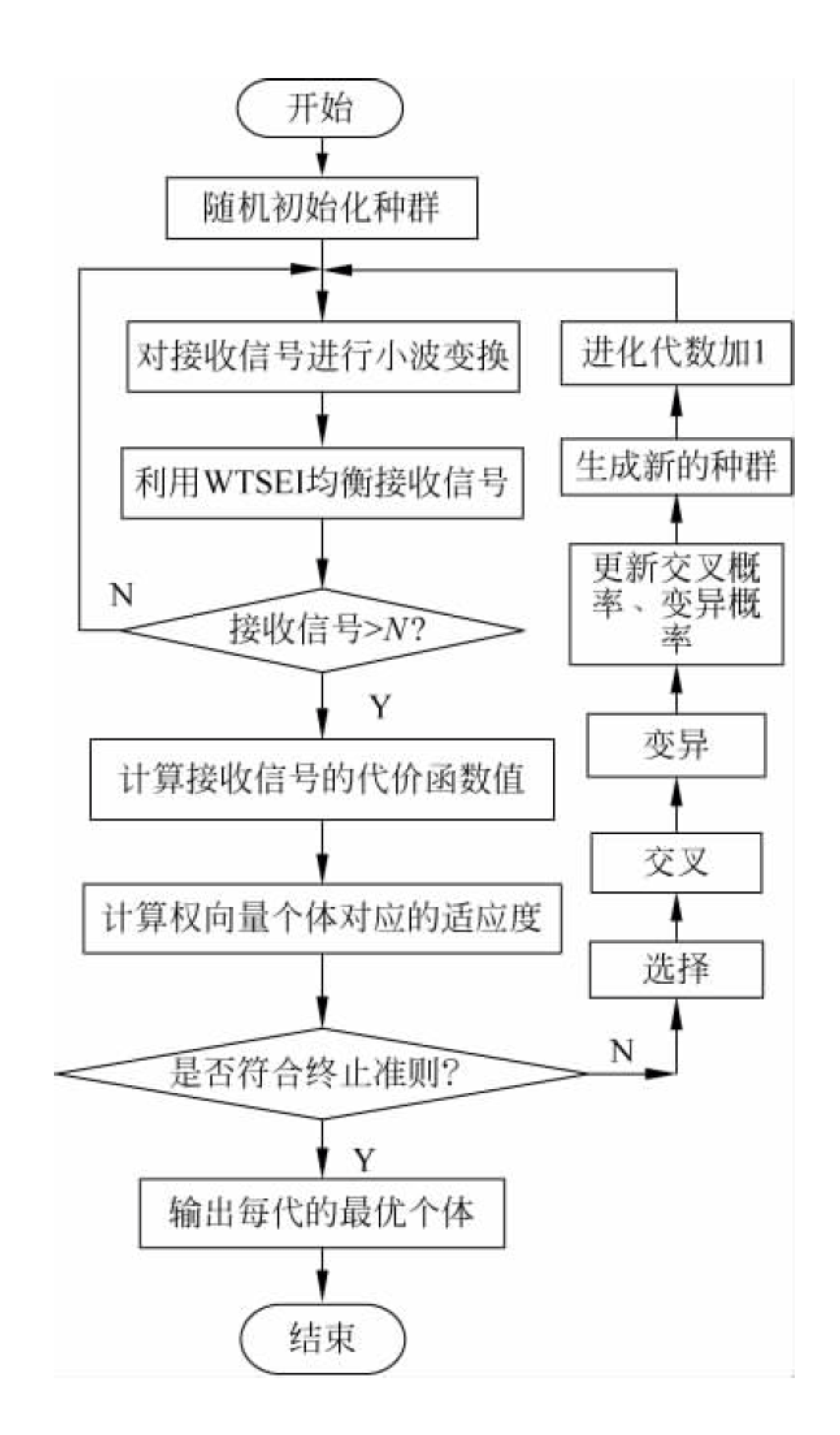



图 2.10　自适应遗传优化算法流程

2. 算法仿真

为了检验 AGA-WTSEI 算法的有效性，以 WTSEI 算法和基于遗传算法优化的正交小波超指数迭代常模盲均衡算法(GA-WTSEI)为比较对象，进行仿真实验。

**【实验 2.3】** 当混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$[10]时；发射信号为 4QAM，均衡器权长为 32，信噪比为 25dB，功率初始值设置为 4，种群规模 100；GA 的交叉、变异概率为 0.9、0.1；AGA 的初始交叉、变异概率为 0.9、0.1，AGA 的提高后的交叉、变异概率为 0.6、0.01，最大进化代数为 50；WTSEI 算法采用第 16 个抽头初始化为 1，其他

参数设置，如表 2.2 所示。当为两径水声信道 $\boldsymbol{h}=[-0.35 \quad 0 \quad 0 \quad 1]$[10] 时，发射信号为 8PSK，均衡器权长为 32，信噪比为 25dB，功率初始值设置为 4，种群规模为 100；GA、AGA 的初始交叉、变异概率分别为 0.7、0.9、1/32、0.1；AGA 的提高后的交叉、变异概率为 0.6、0.01，最大进化代数为 40；WTSEI 算法采用第 20 个抽头初始化为 1，其他参数设置如表 2.3 所示。

表 2.2　仿真参数设置

| 算　　法 | 仿真步长 | 迭代步长 | 小波 | 分解层数 | $\beta$ 取值 |
|---|---|---|---|---|---|
| WTSEI | 0.006 | 0.02 | DB4 | 3 | 0.99 |
| GA-WTSEI | 0.008 | 0.02 | DB4 | 3 | 0.9999 |
| AGA-WTSEI | 0.008 | 0.02 | DB4 | 3 | 0.9999 |

表 2.3　仿真参数设置

| 算　　法 | 仿真步长 | 迭代步长 | 小波 | 分解层数 | $\beta$ 取值 |
|---|---|---|---|---|---|
| WTSEI | 0.015 | 0.02 | DB2 | 3 | 0.99 |
| GA-WTSEI | 0.007 | 0.02 | DB2 | 3 | 0.99 |
| AGA-WTSEI | 0.007 | 0.02 | DB2 | 3 | 0.99 |

在保证星座图完全清晰张开的前提下，500 次蒙特卡罗仿真结果如图 2.11 所示。

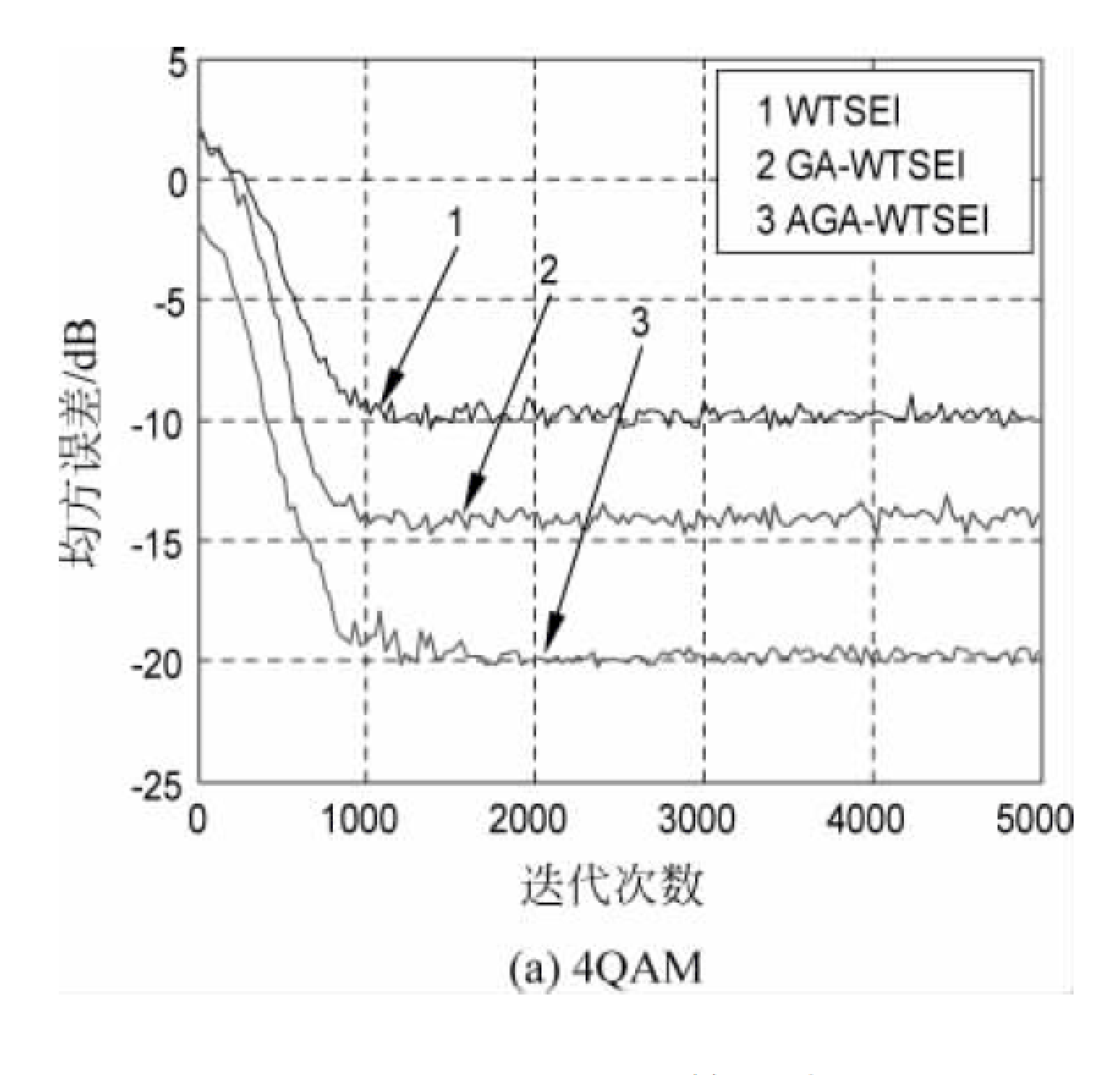
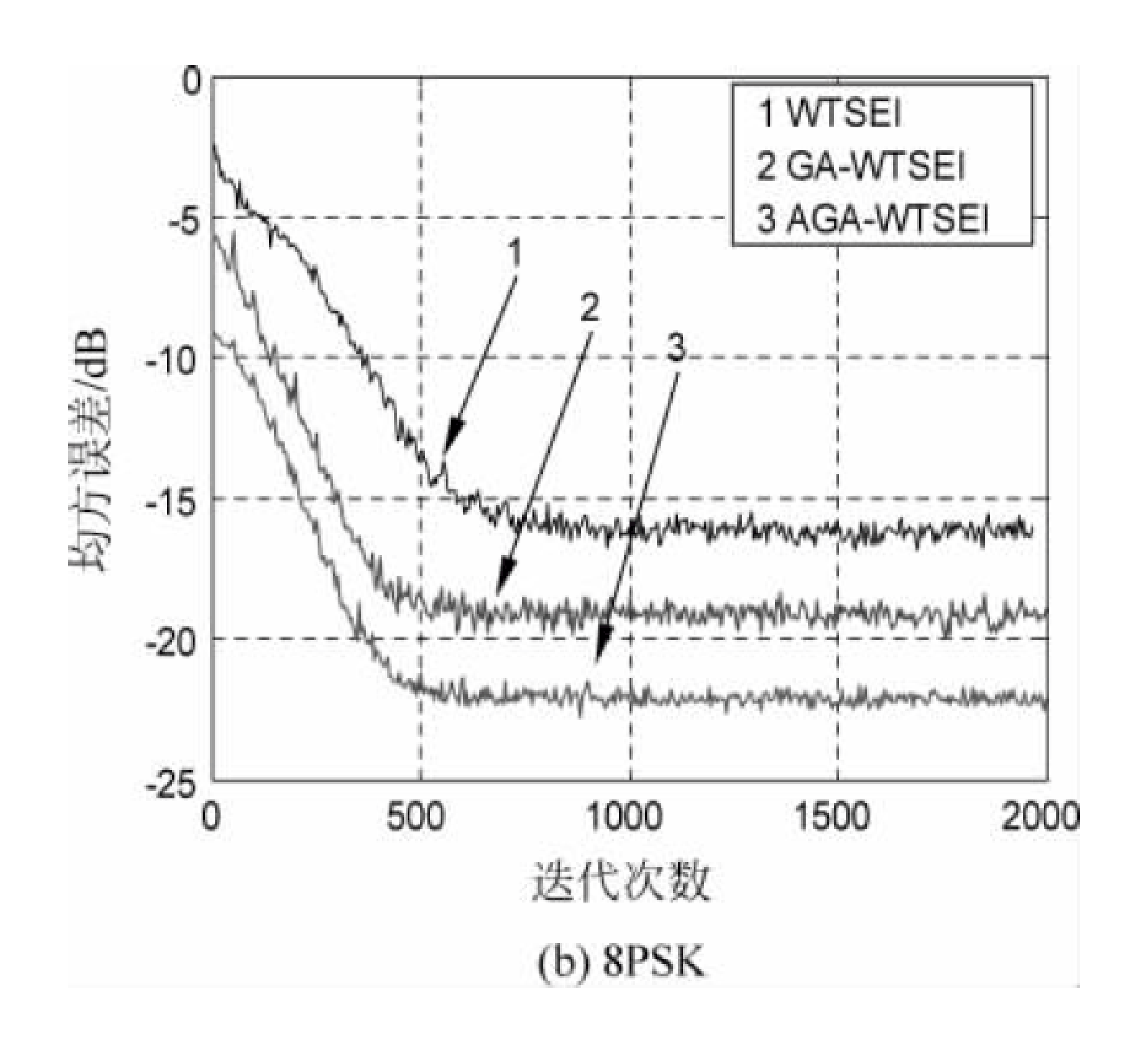



图 2.11　均方误差曲线



(b) 8PSK

图 2.11 （续）

图 2. 11(a)表明，在稳态误差上，与 GA-WTSEI 算法相比，AGA-WTSEI 算法减小了近 5. 5dB；与 WTSEI 算法相比，减小了近 10dB；在收敛速度上，三者基本差不多。图 2. 11(b)表明，在稳态误差上，与 GA-WTSEI 算法相比，AGA-WTSEI 算法减少了近 3dB；与 WTSEI 算法相比，减小了 6. 5dB；在收敛速度上，AGA-WTSEI 算法与 GA-WTSEI 算法基本相同，但比 WTSEI 算法快了 250 步。

## 2.5 基于模拟退火遗传算法优化的正交小波常模盲均衡算法

### 2.5.1 模拟退火遗传算法

模拟退火(simulated annealing algorithm，SA)算法[26-27]是 1982 年 Kirkpatrick 等提出的一种求解大规模组合优化问题的随机方法。模拟退火算法的思想[28]源于统计物理学，是一种模拟固体退火过程并对 Metropolis 算法进行迭代的组合优化算法，它以组合优化问题的求解和统计物理学中的固体退火过程的相似性为基础，并引入 Metropolis 准则。模拟退火算法[29]接受的试探点有使适应度值变大的试探点和使适应度值变

小的试探点，其中后者是以一定的概率被接受，并且此概率会随温度的下降而减小。这种搜索策略有利于减小搜索过程陷入局部最优的可能性，提高了获得全局最优的可靠性。

设组合优化问题的一个解为 $i$，其目标函数为 $J(i)$，它们分别等价于固体的微观状态 $i$ 及其能量 $E_i$；且 $t$ 为控制参数，它等价于固体退火过程中的温度 $T$。在固体模拟退火过程中，它不停地进行"产生新解—判断—接受(舍弃)"的迭代过程，当控制参数 $t$ 每取一个值时，固体模拟退火的迭代过程就会与固体在某一恒定温度下趋于热平衡的过程相对应，即以某一状态为初始状态，通过计算系统的时间演化过程，求出系统最终达到的状态。与此相似，模拟退火算法首先设置所求问题的某个初始解，并对它进行大量的解变换，可以得到给定控制参数下的组合优化问题的相对最优解，然后使控制参数 $t$ 的值逐渐减小，重复执行 Metropolis 算法，最终在控制参数 $t$ 趋于零时，可得组合优化问题的全局最优解。由于固体退火是在慢慢降温条件下使固体在每一温度达到热平衡，并最终趋于固体能量最小的基态，因此模拟退火算法的控制参数 $t$ 在搜索过程中也是缓慢衰减，从而确保算法最终能收敛于组合优化问题的全局最优解[30]。对于模拟退火算法的接受概率可以用统计物理学中的 Metropolis 接受准则来确定。例如，从当前解 $i$ 到新解 $j$ 转移的概率为

$$p_k(i \Rightarrow j) = \begin{cases} 1, & J(i) < J(j) \\ \exp\left(-\dfrac{J(j)-J(i)}{J(j)}\right), & J(i) \geqslant J(j) \end{cases} \tag{2.5.1}$$

在算法执行的初始阶段，控制参数 $t$ 的取值较大，并进行足够多的状态转移后，其值缓慢降低，此过程反复进行，直至满足某个停止准则。因此，模拟退火算法可以看作控制参数 $t$ 不断递减的 Metropolis 算法的迭代过程，具体求解步骤如下。

**步骤 1**：初始化退火温度 $T_0$。

**步骤 2**：随机产生初始解 $X_0$。

**步骤 3**：在温度 $T(k)$时执行：

(1) 在解 $x$ 的可行域中产生新的可行解 $x'$；

(2) 计算新的可行解的评价函数 $J(x')$ 和原来的可行解的评价函数 $J(x)$ 的差值

$$\Delta J = J(x') - J(x)$$

(3) 依概率 $\exp\left(\frac{-\Delta J}{T(k)}\right)$ 接收新的解,若此时的新解达到温度 $T(k)$ 的平衡状态则转步骤 4,否则转步骤 3。

**步骤 4**:以一定方式来降低温度,如 $T(k+1)=\alpha T(k)$,其中 $\alpha\in[0,1]$。

**步骤 5**:若满足收敛准则,则退火过程结束;否则转步骤 3。

通过以上分析可知,模拟退火过程以退火温度 $T$ 来引导求解过程向最小值的优化方向进行,并以概率 $\exp(-\Delta J/T(k))$ 来接收不良解,从而使算法跳出局部极值点,并当初始温度足够高时,退火过程就会足够慢,最终使算法收敛到全局最优解。

### 2.5.2 基于模拟退火遗传算法优化的正交小波常模盲均衡算法

基本遗传算法有较强的全局搜索性[31],但其为单纯性地用适应度值大小来决定所寻解得优劣,当群体中某个体的适应度值很大时,则该个体的基因就会迅速扩散,降低了群体中个体多样性,使搜索停滞不前,陷入早熟现象[32],找不到全局最优解;而且遗传算法的局部搜索能力不强,在进化后期搜索效率较低、收敛速度慢,甚至无法收敛到全局最优解,运算效率不高。因此,将全局搜索能力强的遗传算法与有着优良局部搜索能力的模拟退火算法相结合,得到的相对完善的混合模拟退火遗传算法[33-34],弥补了各自的不足,提高了整体搜索能力与搜索精度,可以更好地解决搜索过程中的局部收敛问题。本节将混合模拟退火遗传算法引入到基于正交小波变换的常模盲均衡算法中,得到了一种基于混合模拟退火遗传算法优化的正交小波常模盲均衡算法(hybrid simulated annealing genetic algorithm based WTCMA,SA-GA-WTCMA)[11,18,19],如图 2.12 所示。它的主体流程是遗传算法运算流程,在主体流程中融入了模拟退火机制,利用它来调整优化群体,从而优化均衡器权向量,其优化流程如图 2.13 所示。

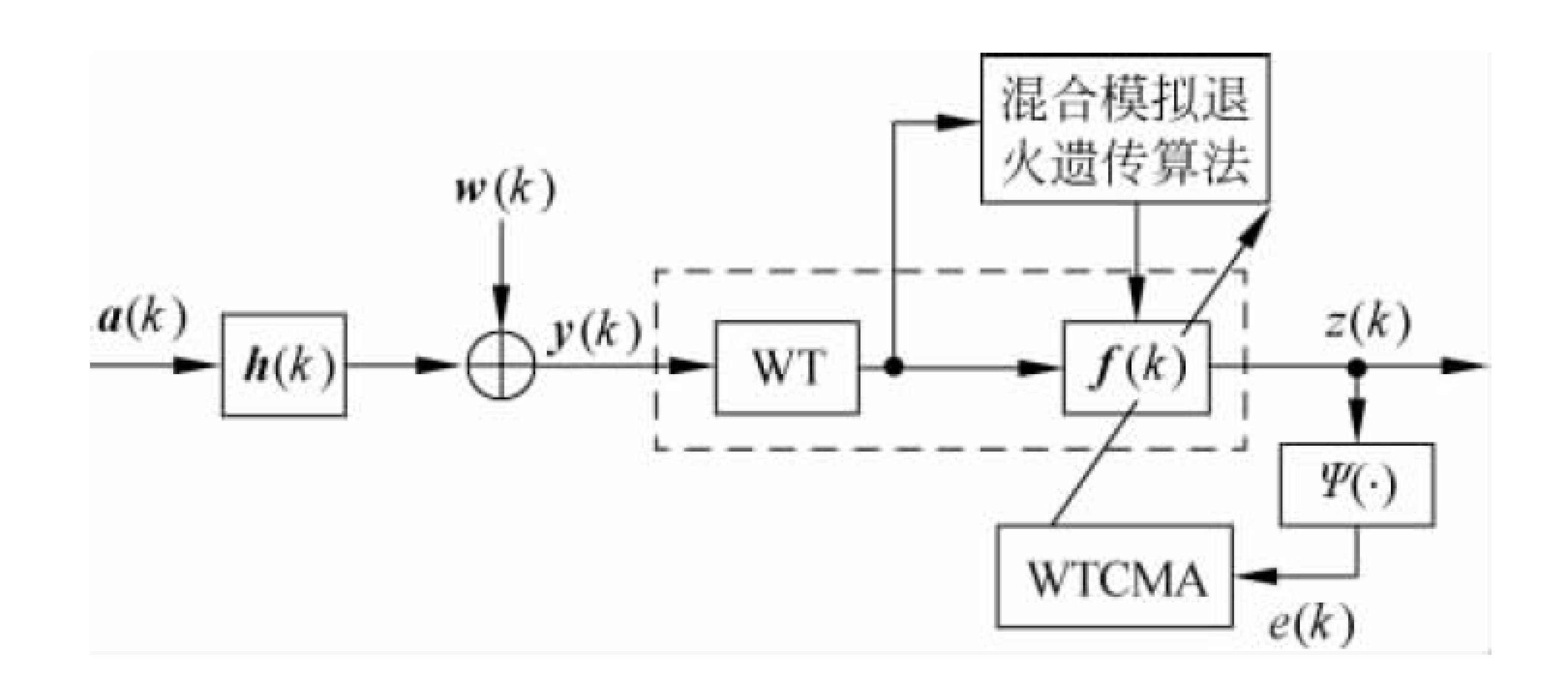



图 2.12 SA-GA-WTCMA 原理

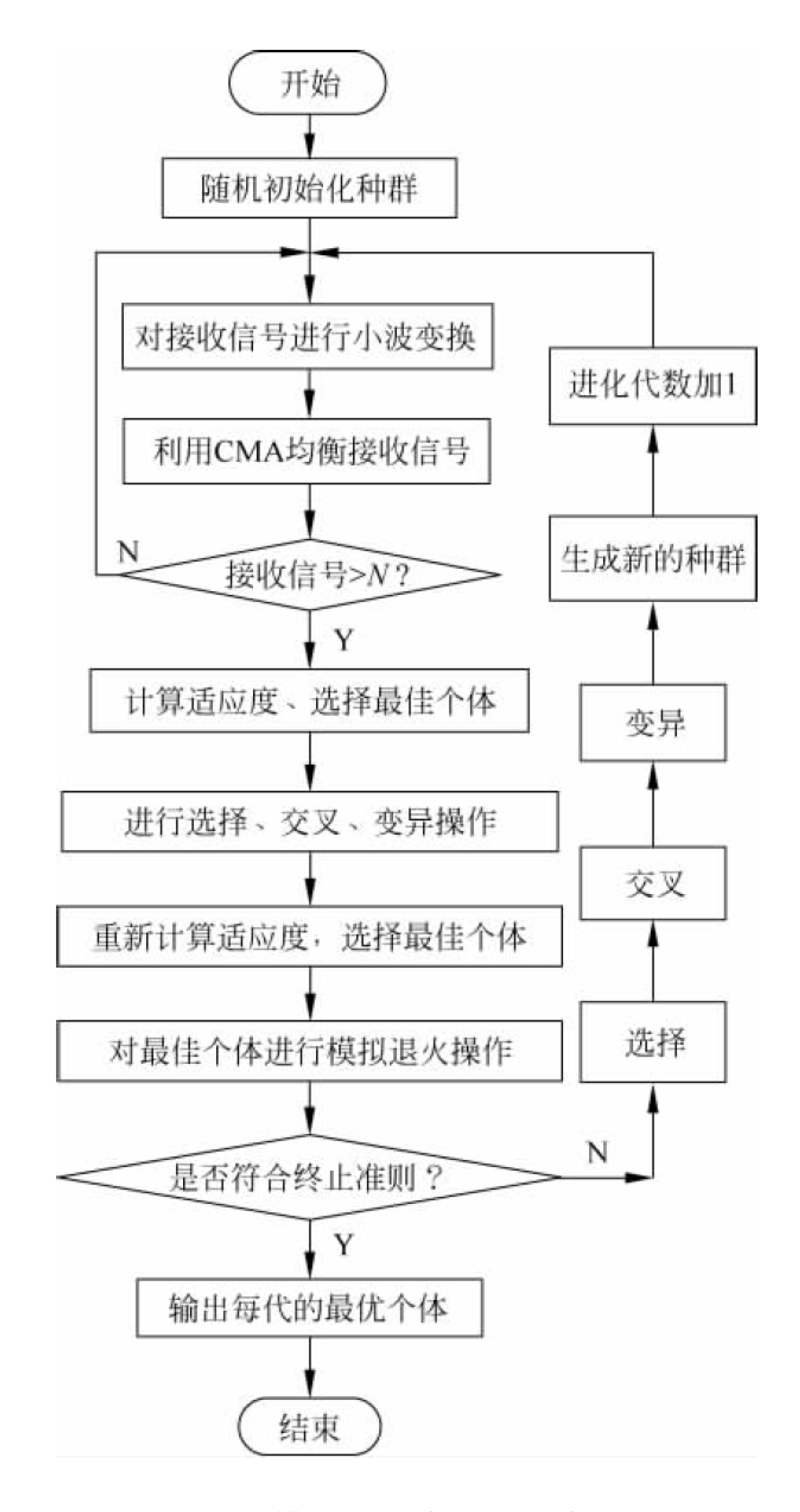



图 2.13 模拟退火遗传优化算法流程

混合模拟退火遗传算法是对经过遗传操作所产生的一组新个体中的最佳个体进行模拟退火操作，将遗传操作前的群体中最佳个体作为原解，将遗传操作后的群体中最佳个体作为新解，采用 Metropolis 准则来弃留新解。在算法优化的每一代，如果这个新解使适应度值增大，那么它就被接受，否则要以指数概率形式来决定它是否被接受。接受新解的概率公式为[33]

$$p(T(k+1)) = \begin{cases} 1, & \text{Fit}(k+1) < \text{Fit}(k) \\ \min\left[1, \exp\left(-\dfrac{\text{Fit}(k+1)-\text{Fit}(k)}{\text{Fit}(k+1)}\right)\right] > \text{randon}(0,1), & \text{Fit}(k+1) \geqslant \text{Fit}(k) \end{cases} \tag{2.5.2}$$

式中，$\text{Fit}(k+1)$为新解的适应度值；$\text{Fit}(k)$为原解的适应度值；$p(T(k+1))$为温度 $T(k+1)$下的接收概率；$T(k+1)$的计算公式为

$$T(k+1) = \alpha \cdot T(k) \tag{2.5.3}$$

式中，$\alpha$ 为温度冷却系数。

### 2.5.3　算法仿真

为了检验 SA-GA-WTCMA 的有效性，以 WTCMA 和 GA-WTCMA 为比较对象，进行仿真实验。

**【实验 2.4】** 当为混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$时，发射信号为 8PSK，均衡器权长为 32，信噪比为 25dB，功率初始值设置为 4，种群规模为 100，交叉概率为 0.7，变异概率为 1/32，最大进化代数为 100，初始温度 $T=100$，温度冷却参数 $k=0.98$，WTCMA 采用第 3 个抽头初始化为 1，其他参数设置如表 2.4 所示。

当为最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$时，发射信号为 16QAM 信号；信噪比为 25dB；均衡器权长为 32；功率初始值设置为 4，种群规模为 100，交叉概率为 0.7，变异概率为 1/32，最大进

化代数为 160。初始温度 $T=100$，温度冷却参数 $k=0.98$，WTCMA 采用第 5 个抽头初始化为 1，其他参数设置如表 2.5 所示。

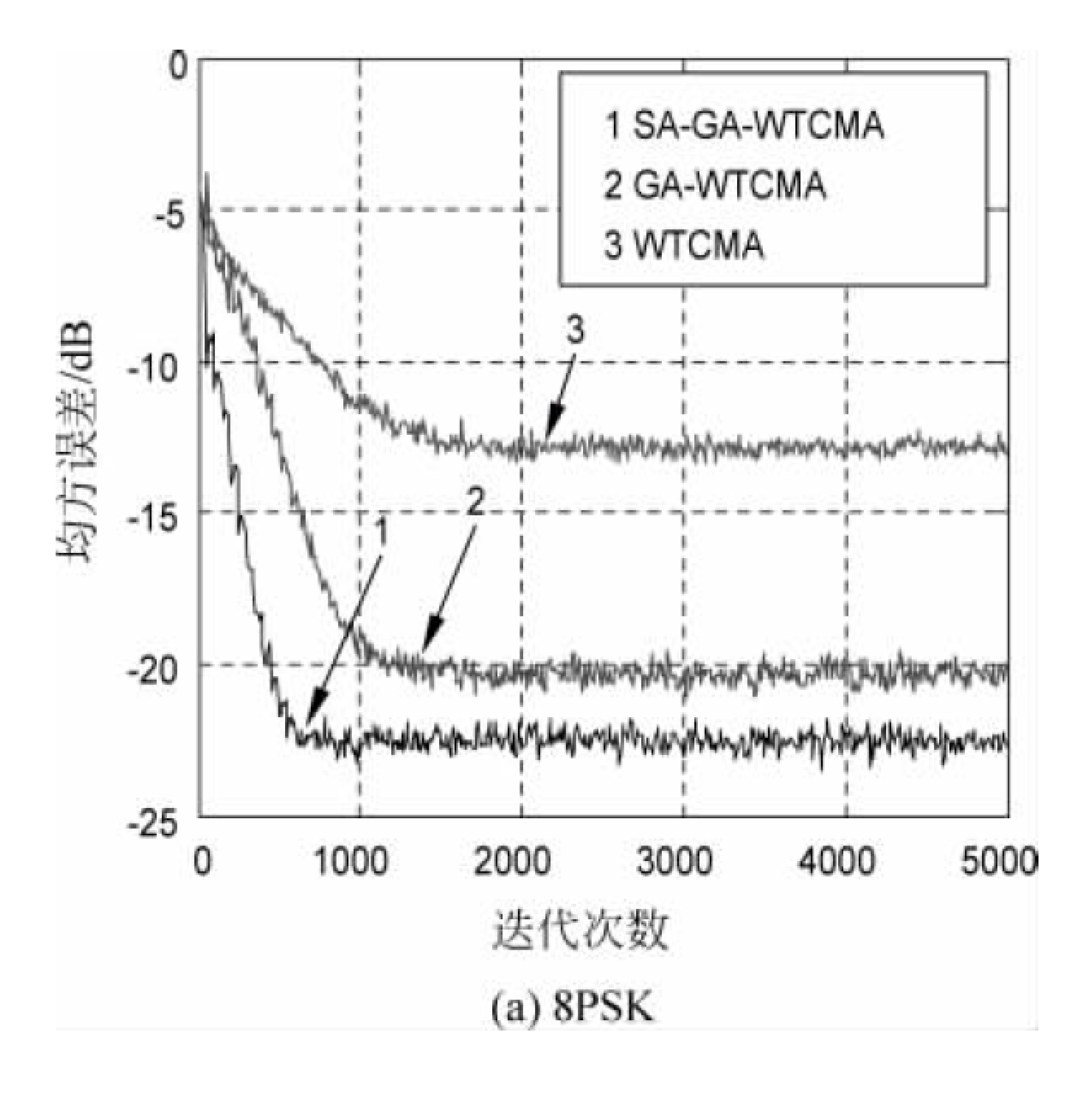
表 2.4　仿真参数设置

| 算　　法 | 仿真步长 | 小波 | 分解层数 | $\beta$ 取值 |
|---|---|---|---|---|
| WTCMA | 0.01 | DB1 | 2 | 0.9999 |
| GA-WTCMA | 0.005 | DB1 | 2 | 0.9999 |
| SA-GA-WTCMA | 0.005 | DB1 | 2 | 0.95 |

表 2.5　仿真参数设置

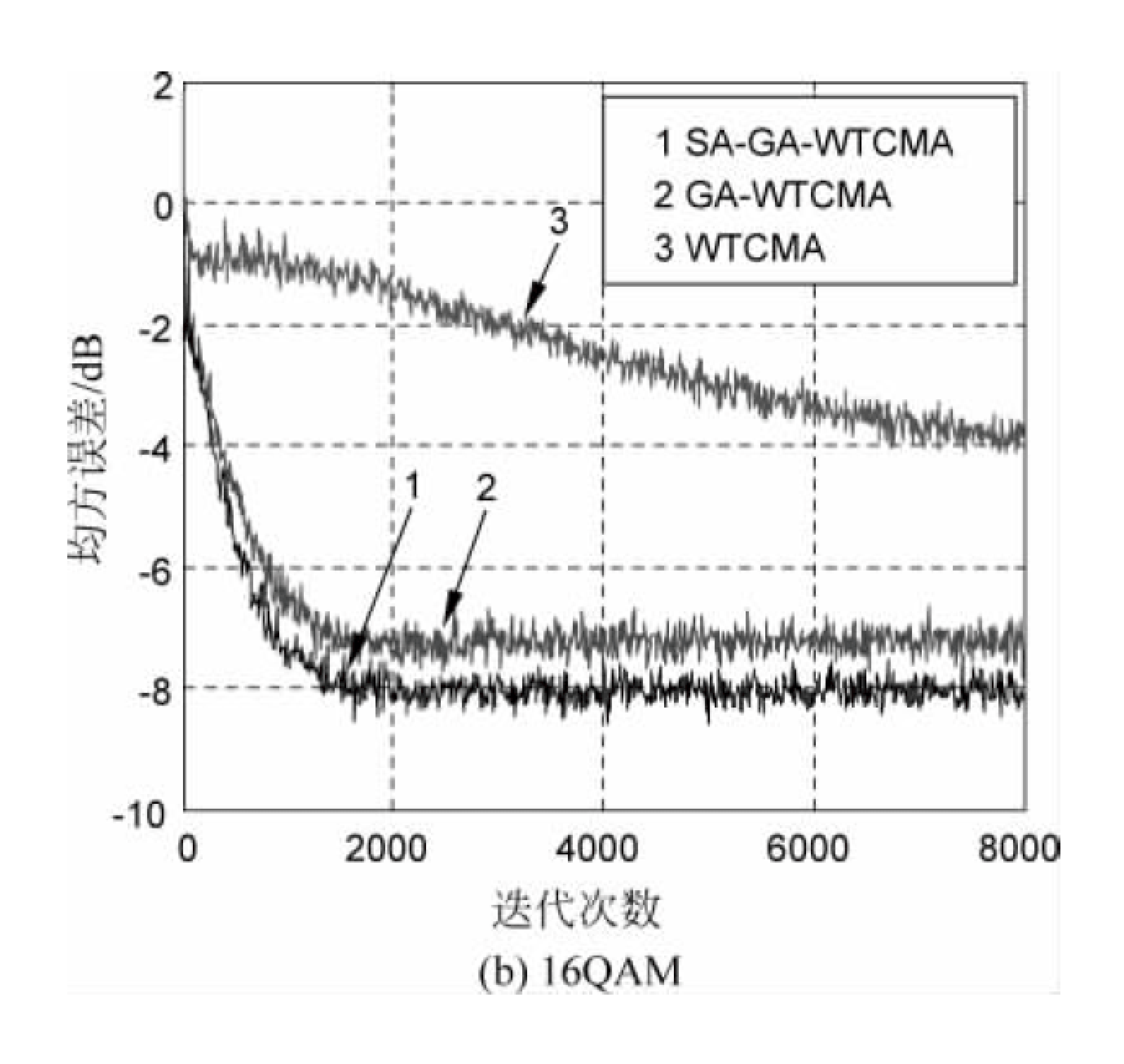
| 算　　法 | 仿真步长 | 小波 | 分解层数 | $\beta$ 取值 |
|---|---|---|---|---|
| WTCMA | 0.000 25 | DB2 | 3 | 0.9999 |
| GA-WTCMA | 0.0002 | DB2 | 3 | 0.9999 |
| SA-GA-WTCMA | 0.000 09 | DB2 | 3 | 0.9 |

在保证星座图完全清晰张开的前提下，500 次蒙特卡罗仿真结果如图 2.14 所示。



(a) 8PSK

图 2.14　仿真结果



(b) 16QAM

图 2.14 （续）

图 2.14(a)表明，在收敛速度上，SA-GA-WTCMA 比 GA-WTCMA 大约快了 500 步，比 WTCMA 快了 1000 步；在稳态误差上，SA-GA-WTCMA 与 GA-WTCMA 相比，减小了近 2.5dB，与 WTCMA 相比，减小了近 10dB。图 2.14(b)表明，在收敛速度上，SA-GA-WTCMA 与 GA-WTCMA 基本相同，但比 WTCMA 快了近 5000 步；在稳态误差上，SA-GA-WTCMA 与 GA-WTCMA 相比，减少了近 1dB，与 WTCMA 算法相比，减小了 4dB。

## 2.6 基于改进混合遗传算法优化的正交小波常模盲均衡算法

传统的遗传算法运行效率不高、局部寻优能力差且易陷入早熟收敛。因此本节先对遗传算法的编码方法、种群初始化、选择算子及交叉算子进行改进后与正交小波变换共同应用到盲均衡算法中，利用正交小波变换的解相关性及遗传算法的全局收敛性和计算速度与精度上的优势，得到了一种基于混合遗传算法优化的正交小波常模盲均衡算法[36-38]。该算法的收敛速度快、均方误差小并能得到全局最优解。

### 2.6.1 混合遗传算法及其改进

#### 1. 混合遗传算法

基本遗传算法从概率的意义上能以随机方式寻求问题的最优解，但也容易产生早熟现象，局部寻优能力较差，并且求解效果比知识型启发式算法差而且不能保证一定能够找到问题的最优解。另外，遗传算法也无法避免多次搜索同一个可行解，这也是影响遗传算法运行效率的一个因素。

另一方面，梯度法、爬山法、模拟退火算法以及列表寻优法等一些优化算法却具有很强的局部搜索能力，而含有与问题相关知识的启发式算法的运行效率也比较高。

将基本遗传算法与问题的特有知识集成到一起所构成的混合遗传算法，主要特点如下。

(1) 引入了局部搜索方法

对群体中各个个体进行局部搜索，从而能够找出个体在当前环境下所对应的局部最优解，这样能达到改善群体总体性能的目的。

(2) 加入了编码变换操作

进行遗传操作时，对种群中的个体进行一定的编码变换；当进入局部搜索方式时，通过解码进行局部最优解的搜索，再将搜索到的优良个体通过编码进入下一代的遗传进化操作。

基本混合遗传算法的框架，如图 2.15 所示。

#### 2. 改进遗传算法操作过程

遗传算法中的五个基本要素：参数编码、初始种群设定、适应度函数设计、遗传算子设计和控制参数设定等，对遗传算法性能有重要影响。针对传统遗传算法计算效率较低、易陷入早熟收敛等缺点，本节采用改进的种群初始化、编码方法、选择算子及交叉算子，以提高计算效率和收敛速度。

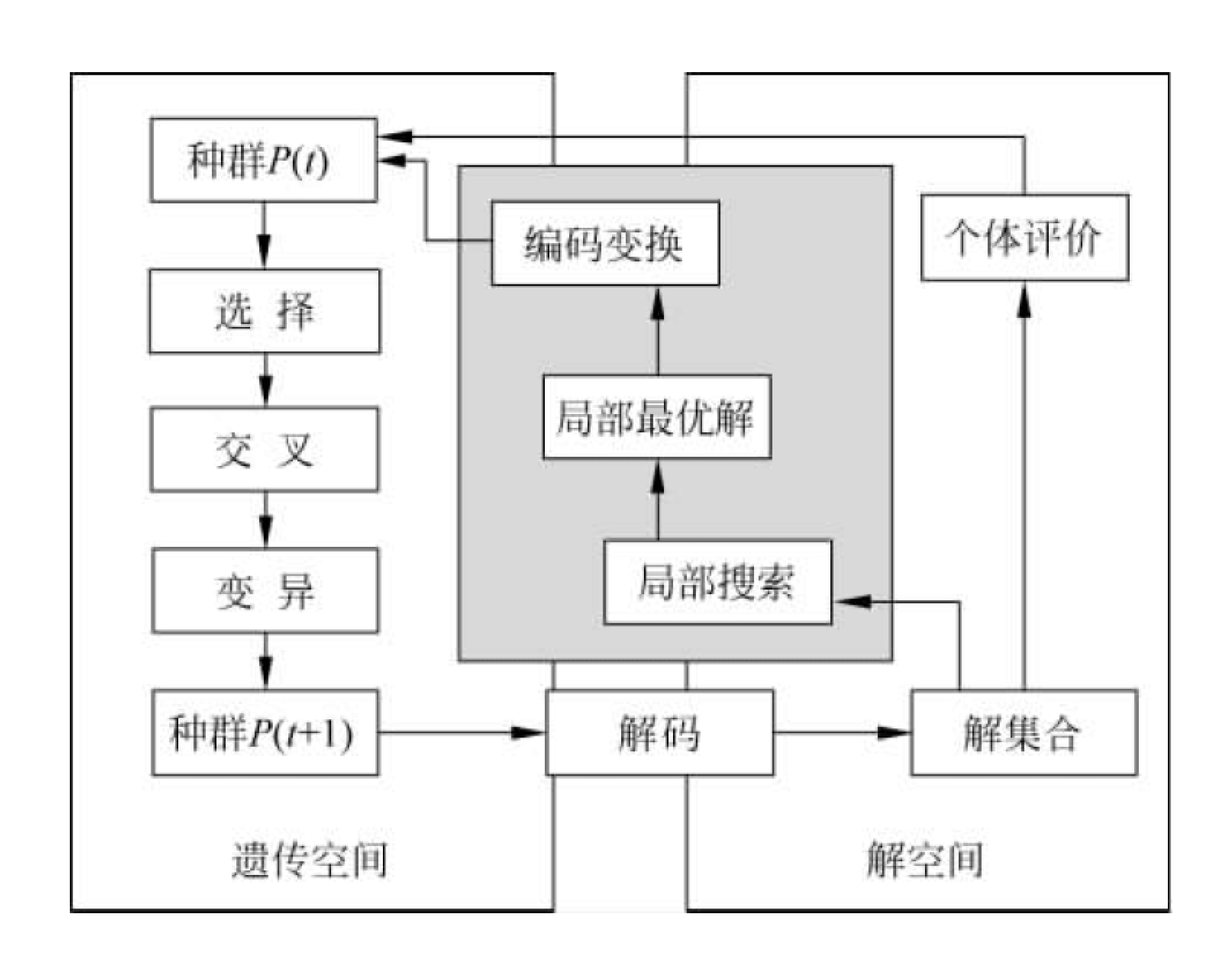



图 2.15 混合遗传算法的基本构成框架

(1) 种群初始化

要得到全局最优解,初始种群在解空间应尽量分散。这样可使初始种群在解空间群体分布均匀,增加获取全局最优解的可能。本节采用小区间生成法产生初始种群,其原理为:首先根据所给出的问题构造均匀数组,然后产生初始种群。初始种群的产生过程如下:

- 将解空间划分为 $S$ 个子空间;
- 量化每个子空间,运用均匀数组选择 $M$ 个染色体;
- 从 $M\times S$ 个染色体中,选择适应度值最大的 $N$ 个个体作为初始群体。

另外,初始群体的各个个体之间应保持一定的距离,这样不仅可以保证初始群体在解空间均匀分布,而且能使随机产生的各个个体间有较明显的差别,使它们能均匀分布在解空间中,从而增加获取全局最优解的可能。

(2) 编码方法

遗传算法的主要数据结构是数字位串,常用的是二进制。当遗传算法应用于连续函数优化计算时,位串长度和编码方式对计算精度和群体中个体之间的距离具有决定性影响,并直接影响全局极值的求解。但二进制编

码有以下两个缺点：首先，二进制编码存在着连续函数离散化时的映射误差，个体编码串的长度较短时，可能达不到精度的要求，而个体编码串的长度较大时，虽然能提高编码精度，但却会使遗传算法的搜索空间急剧扩大；其次，它不能直接反映出所求问题的本身结构特征，这样也就不便于开发针对问题的专门知识的遗传运算算子，很难满足积木块编码原则。

因此本节采用混合式编码，在群体中每个个体采用实数型编码方式，只有在个体被确定进行交叉运算和变异运算时才能进行二进制编码，运算结束后产生的子代进行解码才能进入新的子代群体。采用混合式编码的优点可以避免编码有限字长对精度的影响。

（3）选择算子

在遗传算法中，通过对个体进行交叉、变异等遗传操作而不断产生新的个体，虽然随着群体的进化过程会产生越来越多的优良个体，但由于选择、交叉、变异等操作的随机性，它们也有可能破坏当前群体中适应度值最大的个体，降低群体的平均适应度值，并且对遗传算法的运行效率、收敛性都有不利的影响。因此，适应度值最大的个体要尽量保留到下一代群体中。为达到这个目的，可以使用精英保留策略进化模型来进行优胜劣汰操作，即当前群体中适应度值最大的个体不参与交叉运算和变异运算，而是用它来替换掉本代群体中经过交叉、变异等操作后所产生的适应度函数值最小的个体。

精英保留策略的具体操作过程为：

- 确定群体中适应度值最大的个体和适应度值最小的个体。
- 群体中最优个体的适应度值比总的迄今为止的最优个体的适应度值还要高，则以当前群体中的最优个体作为新的迄今为止的最优个体。
- 用迄今为止最优的个体替换掉当前群体中的最差个体。

精英保留策略可是为选择操作的一部分，该策略的实施可保证迄今为止所得到的最佳个体不会被交叉、变异等遗传运算所破坏，它是遗传算法收敛性的一个重要保证条件。另一方面，它也容易使某个局部最佳个体不易被淘汰反而快速扩散，从而使算法的全局搜索能力不强。所以，该方法一般

要与其他一些选择操作方法配合起来使用，方可有良好的结果。

因此，本节采用轮盘赌选择和精英保留策略相结合的混合选择算子，将精英个体直接复制到下一代，其余个体仍采用轮盘赌选择。

轮盘赌选择是基于个体适应度值的选择机制，而精英保留策略是群体收敛到优化问题最优解的一种基本保障，将这两种方法相结合能够克服算法进化早期早熟收敛和进化后期搜索效率下降的缺点。

(4) 交叉算子

交叉及变异操作能够产生新个体，是遗传算法区别于其他进化算法的重要特征。POX 算子是目前应用较为广泛的一种交叉算子，其主要思想就是让染色体上的部分基因保持原来的位置，剩下的基因采用另一条染色体的排列顺序，从而产生的全都是可行解，能够很好地继承父代的优良特性。因此本节采用改进的 POX 算子，该算子使父代的两条遗传的染色体各自保留优良的不同的基因组合，剩下的基因组合仍然用另一条染色体上的排列顺序。经过这样的改进，使得染色体可以到达的范围更大，产生最优解的机会就变大，使结果容易达到最优解。本节交叉算子的设计如图 2.16 所示。

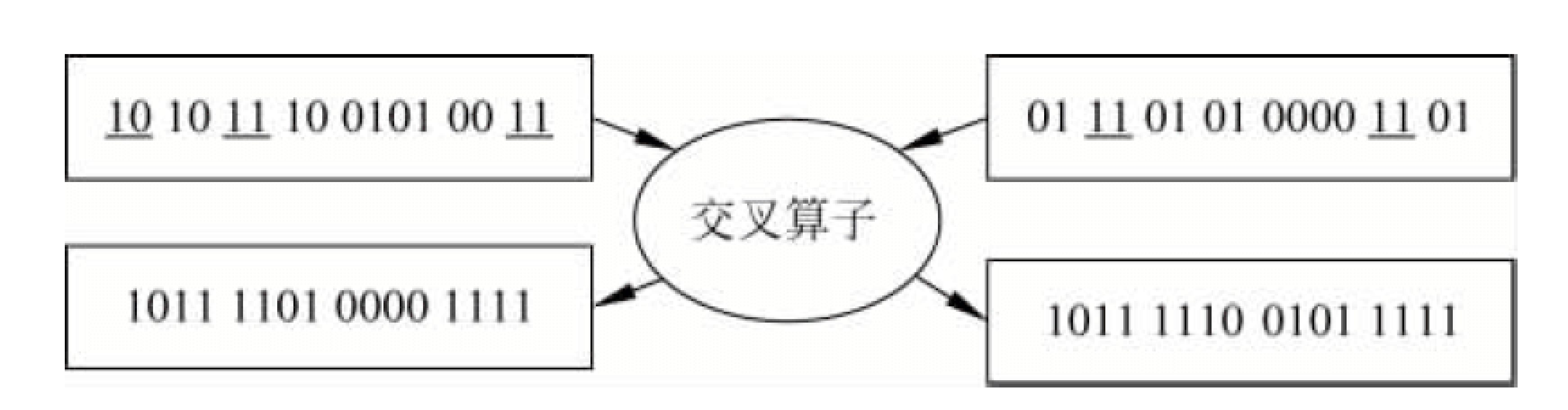



图 2.16　改进的选择算子

(5) 变异算子

交叉运算是产生新个体的主要方法，它决定了遗传算法的全局搜索能力，而变异运算只是产生新个体的辅助方法，也是必不可少的一个步骤，因为它决定了遗传算法的局部搜索能力。交叉运算与变异运算相互配合，共同完成对搜索空间的全局搜索和局部搜索，从而使得遗传算法能够以良好的搜索性能完成最优化问题的寻优过程。

本节采用非均匀变异。均匀变异操作只取某一范围内均匀分布的随机数来替换原有基因值，可使得个体在搜索空间内自由移动，但另一方面，它

却不便于对某一重点区域进行局部搜索。为克服这一不足，本节不用均匀分布的随机数去替换原有的基因值，而是对原有的基因值做一随机扰动，以扰动后的结果作为变异后的新基因值。对每个基因座都以相同的概率进行变异运算之后，相当于整个解向量在解空间中作了一个轻微的变动，其便于对某一重点区域进行局部搜索。

### 2.6.2 基于改进混合遗传优化的正交小波常模盲均衡算法

(1) 算法描述

本节用改进混合遗传算法对正交小波常模盲均衡算法进行优化，得到基于改进混合遗传优化的正交小波常模盲均衡算法(modified hybrid genetic algorithm based WTCMA，MHGA-WTCMA)。其优化流程如图 2.17 所示。

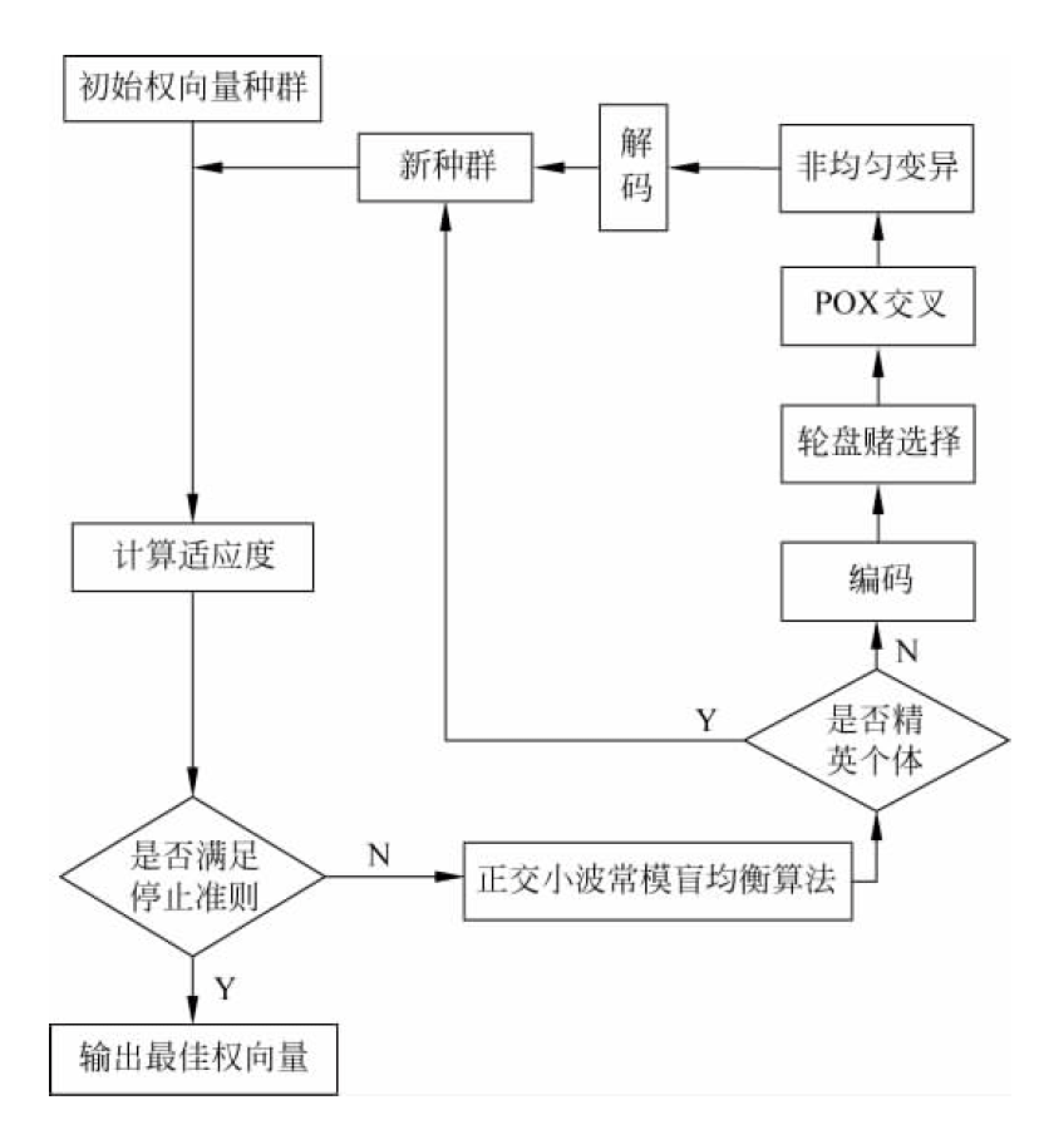



图 2.17 改进混合遗传优化流程

该算法中，将均衡器权向量作为遗传算法的决策变量，设计初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$，其中 $M$ 为个体数量。种群中每一个个体 $\boldsymbol{f}_m(1\leqslant m\leqslant M)$ 对应一个均衡器权向量，而本节由小区间生成法构造初始种群。考虑到盲均衡的特点，权向量每一项的模必定小于 1，因此将搜索范围确定为 [0,1]，将搜索范围分为 10 个子区间，每个子区间内均匀取 2000 个个体，通过适应度函数计算，选择适应度值最大的 10 000 个个体作为初始种群。

设接收信号序列的长度为 $N$，利用时间平均代替统计平均，常数模算法代价函数定义为

$$J(\boldsymbol{f}_m)=\frac{\sum_{i=1}^{N}(R^2-|z_m(i)|^2)^2}{N} \tag{2.6.1}$$

式中，$m$ 表示均衡器权向量个体的序号；$N$ 为每一代接收信号序列的长度；$z_m(i)$ 为均衡器第 $m$ 个权向量个体的输出信号。式(2.6.1)为混合遗传算法的目标函数，求其最小值所得的最优个体就是盲均衡算法权向量的最优解，因此，个体的适应度函数定义为

$$\text{Fit}(\boldsymbol{f}_m)=\frac{b}{J(\boldsymbol{f}_m)} \tag{2.6.2}$$

式中，$b$ 为比例系数。

将前一代种群通过正交小波常模盲均衡算法搜索该种群中的精英个体并直接复制到新种群中，其余个体进入遗传算法操作过程。由于经过正交小波常模盲均衡算法搜索产生的精英个体是种群中适应度值最大的个体，其继承了父代的优良品质和基因结构，因此不会被选择、交叉和变异操作所丢失和破坏。正交小波常模盲均衡算法在混合遗传算法中起到了局部搜索的作用，处理了小范围搜索和搜索加速的问题。

而其余个体进行二进制编码以及遗传算法的轮盘赌选择、POX 交叉和非均匀变异操作，经过解码再进入新种群中。如果不符合遗传算法的终止要求，将进行下一次混合遗传算法优化，直到满足条件为止。基于混合遗传算法优化的正交小波常模盲均衡算法充分利用了遗传算法和正交小波常模盲均衡算法各自的优点，从而得到更好的盲均衡效果。

(2) 算法仿真

为了检验 MHGA-WTCMA 的性能,以 CMA 和 WTCMA 为比较对象,进行仿真实验。

**【实验 2.5】** 信道 $\boldsymbol{h}=[0.3132,-0.104,0.8908,0.3134]$,发射信号为 16QAM,信噪比为 15dB,均衡器长度为 16,MHGA-WTCMA 训练样本个数 $N=10\ 000$。

CMA 初始权向量第 4 个抽头取 1,其他都取 0,步长取 0.000 02;WTCMA 第 4 个抽头取 1,其他都取 0,步长取 0.000 005。

WTCMA 和 MHGA-WTCMA 采用 Db4 小波,分解层数为 2 层,$\beta$ 取值为 0.99,功率初始化值为 10;

MHGA-WTCMA 种群规模取 20,交叉概率取 0.7,变异概率取 1/16,终止条件为进化至第 10 代。仿真结果如图 2.18 所示。

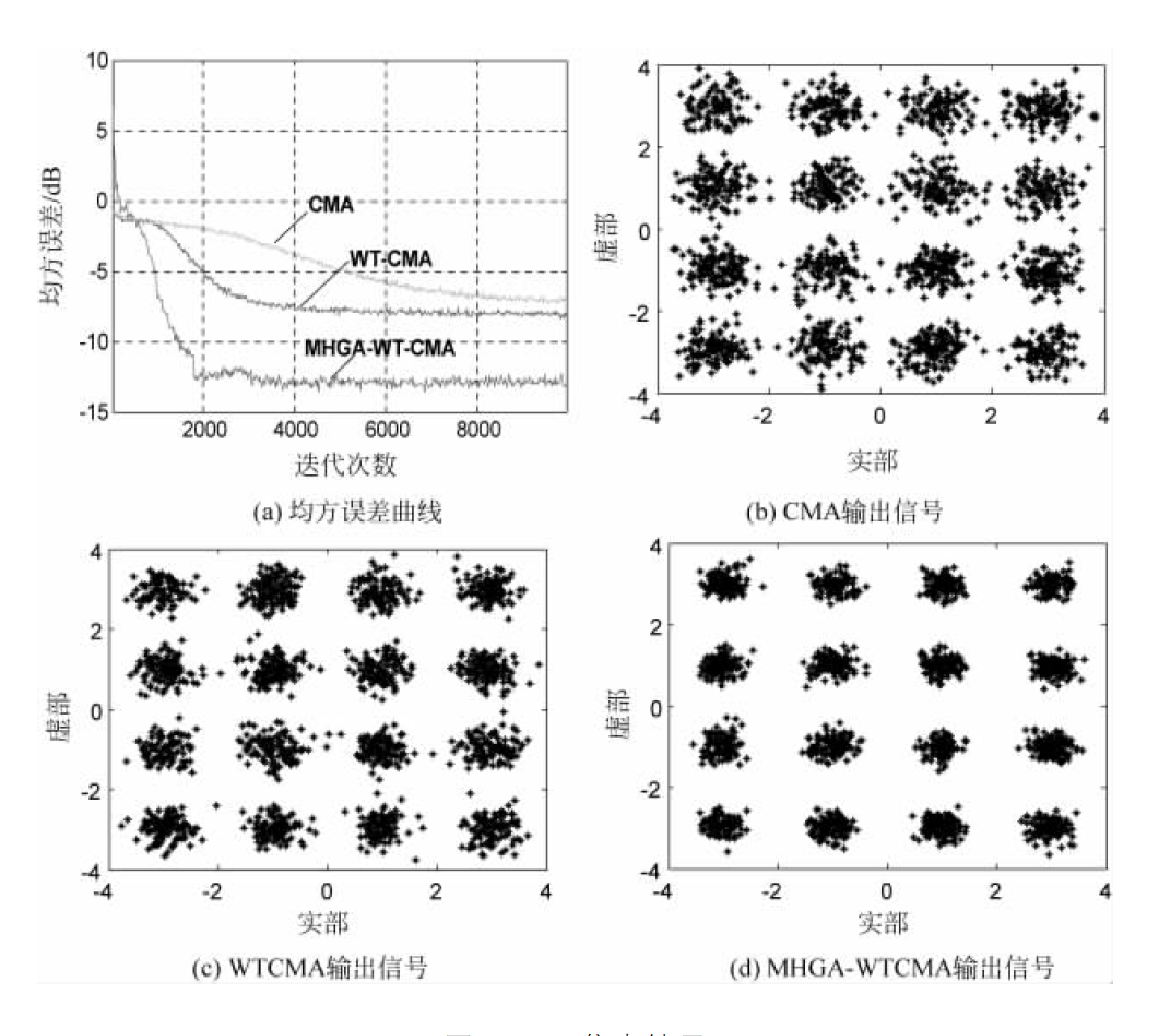



图 2.18 仿真结果

图 2.18(a)表明,MHGA-WTCMA 的收敛速度较 CMA 和 WTCMA 分别提高了近 5000 步和 2000 步,且均方误差也分别减小了近 6dB 和 5dB。

图 2.18(b)～(d)表明,MHGA-WTCMA 输出星座图较 CMA 和 WTCMA 图更加清晰。上述结果表明,经过改进的混合遗传算法优化后的正交小波常模盲均衡算法,可以有效地加快收敛速度并减小均方误差,性能得到明显地提高。

## 参考文献


[1] Bethke A D. Comparison of genetic algorithms and gradient-based optimizers on parallel processors: efficiency of use of processing capacity(Technical Report No. 197)[R]. Ann Arbor: University of Michigan,Login of Computers Group, 1976.

[2] Booker L B, Goldberg D E, Holland J H. Classifier systems and genetic algorithm[J]. Artificial Intelligence,1989,40: 235-282.

[3] Bertoni A, Dorigo M. Implicit parallelism in genetic algorithms[J]. Artificial Intelligence,1993,61: 307-314.

[4] 陈国良,王熙法,庄镇泉. 遗传算法及其应用[M]. 北京: 人民邮电出版社,1996.

[5] 罗文辉. 遗传算法在神经网络优化中的应用[J]. 控制工程,2003,10(5): 401-403.

[6] 周激流. 遗传算法理论及其在水问题中的研究[D]. 成都: 四川大学,2000.

[7] 周明,孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社,1999.

[8] 王小平,曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社,2002.

[9] Philip Schniter, C Richard Johnson. Dithered signed-error CMA: robust, efficient blind adaptive equalization [J]. IEEE Transactions on Signal Computationally Processing,1999,47(6): 1592-1603.

[10] 王峰. 基于高阶统计量的水声信道盲均衡算法研究[D]. 西安: 西北工业大学,2003.

[11] 廖娟. 基于遗传算法优化的正交小波盲均衡算法[D]. 淮南: 安徽理工大学,2011.

[12] 樊康. 基于遗传优化的正交小波盲均衡算法[D]. 南京：南京信息工程大学，2011.

[13] 王卫. 基于改进遗传优化的正交小波盲均衡算法[D]. 南京：南京信息工程大学，2012.

[14] 刘锋，葛临东，吴业进，等. 一种基于实数编码遗传算法的常数模盲均衡算法[J]. 电讯技术，2008，10(48)：5-8.

[15] 韩迎鸽，郭业才. 引入动量项的正交小波变换盲均衡算法[J]. 系统仿真学报. 2008，20(6)：1559～1562.

[16] 韩迎鸽，郭业才，吴造林，等. 基于正交小波变换的多模盲均衡器设计与算法仿真研究[J]. 仪器仪表学报，2008，29(7)：1441-1445.

[17] 郭业才，杨超. 基于正交小波变换的超指数迭代盲均衡器设计与仿真[J]. 系统仿真学报. 2009，21(20)：6556-6559.

[18] 廖娟，郭业才，刘振兴，等. 基于遗传优化的正交小波分数间隔盲均衡算法[J]. 兵工学报，2011，32(3)：268-274.

[19] LIAO Juan, GUO Ye-cai, JI Tong-ying. An orthogonal wavelet transform fractionally spaced blind equalization algorithm based on the optimization of genetic algorithm[J]. Journal of China Ordnance，2011，7(2)：65-72.

[20] Scrinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Trans on Systems，Man and Cybernetics，1994，24(4)：656-667.

[21] 房磊，张焕春，经亚枝. 一种基于 FPGA 的自适应遗传算法[J]. 电子与信息学报，2005，27(11)：1829-1833.

[22] 王蕾，沈庭芝，招扬. 一种改进的自适应遗传算法[J]. 系统工程与电子技术，2002，24(5)：75-78.

[23] 马均水，刘忠贵，贾玉兰. 改进遗传算法搜索性能的大变异操作[J]. 控制理论与应用，1998，15(3)：404-408.

[24] Weiguo Liang, Tongtong Li, J K Tugnait. Blind detection of multirate asynchronous CDMA signals using super-exponential methods[J]. IEEE Signal Processing Letters，2005，13(3)：218-221.

[25] 郭业才，杨超. 基于正交小波变换的超指数迭代联合盲均衡算法[J]. 数据采集与处理，2010，25(1)：13-18.

[26] Roya S Nikjoo. A hybrid genetic-simulated annealing algorithm for optimal design of grounding grids with Rods [J]. IEEE Power Transmission Technologies，Asia-Pacific Power and Energy Engineering Conference 2010.

[27] Liu Wan-hui, Tian Shu-jun, Jia Chun-qiang, et al. A hybrid genetic-simulated annealing algorithm for optimization of hydraulic manifold blocks[J]. Journal of Shanghai University, 2008, 12(3): 261-267.

[28] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing [J]. Science, 1983, 220: 671-680.

[29] 吕品，张金芳，鲍冠佰，等. 基于改进模拟退火算法的观察点设置问题研究[J]. 系统仿真学报，2009，21(14): 4328-4330.

[30] S Pai, W K Jenkins, D J Krusienski. Adaptive IIR phase equalizers based on stochastic search algorithms[C]. Conference: Signals, Systems and Computers, 2003: 652-656.

[31] 裴志刚. 模拟退火遗传算法在飞行冲突解脱中的应用[D]. 西安: 西北工业大学，2005.

[32] 寇华，王宝树. 基于遗传算法的 RBF-PLS 方法在辐射源识别中的应用[J]. 电子与信息学报，2007，29(5): 1031-1034.

[33] 邹谊，魏文龙，李斌，等. 多目标量子编码遗传算法[J]. 电子与信息学报，2007，29(11): 2688-2692.

[34] 康立山，等. 模拟退火算法[M]. 北京: 科学出版社，1994.

[35] 于斌，尹成友，黄冶. 基于模拟退火算法校正天线阵列方向性误差[J]. 系统工程与电子技术，2007，29(2): 174～177.

[36] 郭业才，王卫. 基于改进混合遗传的正交小波盲均衡算法[J]. 信号处理，2011，27(7): 1004-1009.

[37] 郭业才，樊康，徐文才，等. 基于混合遗传优化的正交小波变换盲均衡算法[J]. 数据采集与处理，2011，26(5): 503-507.

[38] 樊康，郭业才. 基于小生境遗传优化的正交小波变换盲均衡算法[J]. 微电子学与计算机，2011，28(7): 50-53.

# 第 3 章　基于混沌优化的盲均衡算法

**【内容导语】** 本章从混沌定义入手，在分析非线性动力学系统中的混沌、混沌运动的随机性特征、混沌序列基本特点和混沌优化算法流程基础上，将混沌优化算法引入到正交小波常模盲均衡算法中，对正交小波常模盲均衡算法进行了优化；将混沌优化算法与支持向量机相结合，优化了正交小波加权多模盲均衡算法。

## 3.1　混沌理论

### 3.1.1　非线性动力学系统中的混沌

动力系统是用来描述运动的某种特性随时间变化的数学模型和准则，对系统的状态变量进行分析。系统状态变量按照某一确定性规则随时间变化，变量的个数通常表示为相空间的维数。根据状态变量变化规则在时间上是否连续，通常分为连续动力系统和离散动力系统。

$m$ 维连续动力系统[1]表示为

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}(t)=P(\boldsymbol{X}(t)) \tag{3.1.1}$$

式中，$\boldsymbol{X}(t)$是 $m$ 维欧氏空间中的一个向量，表示在 $t$ 时刻的状态。

离散动力学系统的差分方程为

$$\boldsymbol{X}(k+1)=F(\boldsymbol{X}(k)) \quad k=1,2,3,\cdots \tag{3.1.2}$$

式中，$\boldsymbol{X}(k)=[x_1(k),x_2(k),\cdots,x_m(k)]$是 $\boldsymbol{X}(t)$离散向量，$x_m(k)$为 $k$ 时刻第 $m$ 维状态变量，$F$ 表示映射函数。

当动力系统中的状态向量经过一段时间演化后，最终收缩到 $m$ 维相空间中若干个维数低于 $m$ 的有限范围上，该有限空间被称为吸引子。概括地说，吸引子就是一个不变集。若 $X$ 的开邻域 $P$ 满足当 $t\to\infty$时，$R_t(N)\to R_t(\Psi)$，同时 $\Psi$ 不可再分割成为更小的集合 $\Psi_1,\Psi_2,\cdots$，使得 $R_t(\Psi_1)\cap R_t(\Psi_2)\neq\phi$，则 $\Psi$ 为吸引子。在一维相空间中，它的吸引子是稳定的不动点；对于二维相空间，吸引子是不动点或简单的闭合曲线；三维空间中的吸引子可以产生不动点、闭合曲线或准吸引子即奇怪吸引子。

对于奇怪吸引子所对应的动力系统，其运动具有无序、随机、不可预测和对初值具有敏感依赖的特点。奇怪吸引子是非线性动力系统中出现混沌运动的关键，因此奇怪吸引子又称为混沌吸引子。

到目前为止，由于混沌现象的复杂性，混沌还缺乏一个通用的标准定义。学者李天岩(Li T. Y.)和 Yorke J. A. 从遍历论中测度理论上给出了混沌定义[23]；后来 Devaney R. L. 从拓扑意义上给出了混沌定义；分述如下。

**【定义 3.1】** 它是基于 Li-Yorke 定理的严格定义。Li-Yorke 定理：设 $F(x)$是$[a,b]$上的连续自映射，若 $F(x)$有 3 个周期点，则对任何正整数 $n$，$F(x)$有 $n$ 个周期点，也就是著名的“周期三意味着混沌”。混沌定义为：闭区间 $I$ 上的连续自映射 $F(x)$(简记为 $F$)，若满足下列 2 条件，则一定出现混沌现象[2]：

(1) $F$ 周期点的周期无上界；

(2) 闭区间 $I$ 上存在不可数子集 $S$，满足：

① 对任意 $x,y\in S$，当 $x\neq y$ 时，有

$$\lim_{n\to\infty}\sup|F^n(x)-F^n(y)|>0$$

② 对任意 $x,y\in S$，有

$$\lim_{n\to\infty}\inf|F^n(x)-F^n(y)|=0$$

③ 对任意 $x,y\in S$，其中 $y$ 是 $F$ 的任一周期点，则有

$$\lim_{n\to\infty}\sup|F^n(x)-F^n(y)|>0$$

**【定义 3.2】** 设 $V$ 为一个集合，$F:V\to V$ 称为在 $V$ 上是混沌的，若：(a)$F$ 对初始条件的敏感依赖性；(b)$F$ 是拓扑传递的(即对任何一对开集 $V_1,V_2\subset V$，存在 $k>0$ 使 $F^k(V_1)\cap V_2\neq\phi$)；(c)周期点在 $V$ 中稠密。

### 3.1.2 混沌运动的随机性特征

混沌的定义 3.2 表明，混沌运动具有对初值的敏感性和确定性，因此混沌信号是有界的、确定性的类随机信号，具有以下几个与随机性现象类似的特征[3]。

#### 1. 时域上的随机性和初值敏感性

与常见的周期信号不同，混沌信号总是呈现出随机混乱的现象，具有类随机信号的特性；但它并不是随机信号，当混沌系统的初始值给定时，其所产生的混沌信号是唯一确定的。混沌信号的随机性现象可以从图 3.1 中看出，该图采用 Logistic 混沌映射函数，即

$$x(k+1)=\mu x(k)[1-x(k)] \tag{3.1.3}$$

式中，$x(k)$为混沌变量。该映射在 $\mu>3.05$ 时处于混沌状态。图 3.1 表明，混沌信号呈现一种非周期的现象。

#### 2. 不可预测性

由混沌的定义可知，混沌运动对初始值具有很强的敏感性，当混沌系统的 Lyapunov 指数为正数时，对于初始值不同的混沌系统，其所产生混沌信

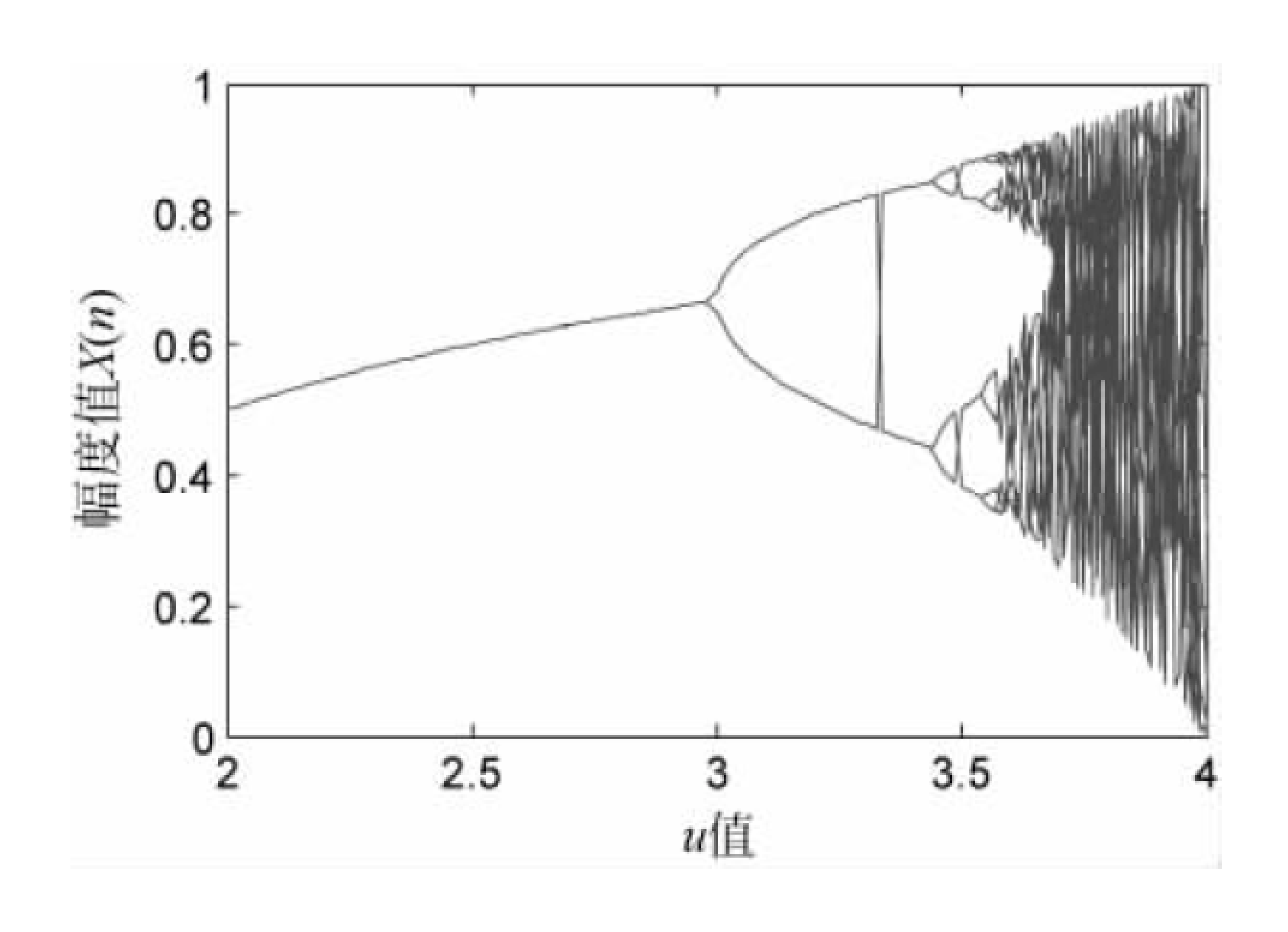



图 3.1　logistic 映射的时域特性

号的模值成指数分离，即当初始条件中产生很小的测量误差，会导致信号的模值成指数分离，使得原本确定的混沌系统失去预测的能力。图 3.2 中，相差很小的两个混沌初始值，在迭代 20 多次后，其模值在逐渐分离，成为两条毫无关系的轨道。

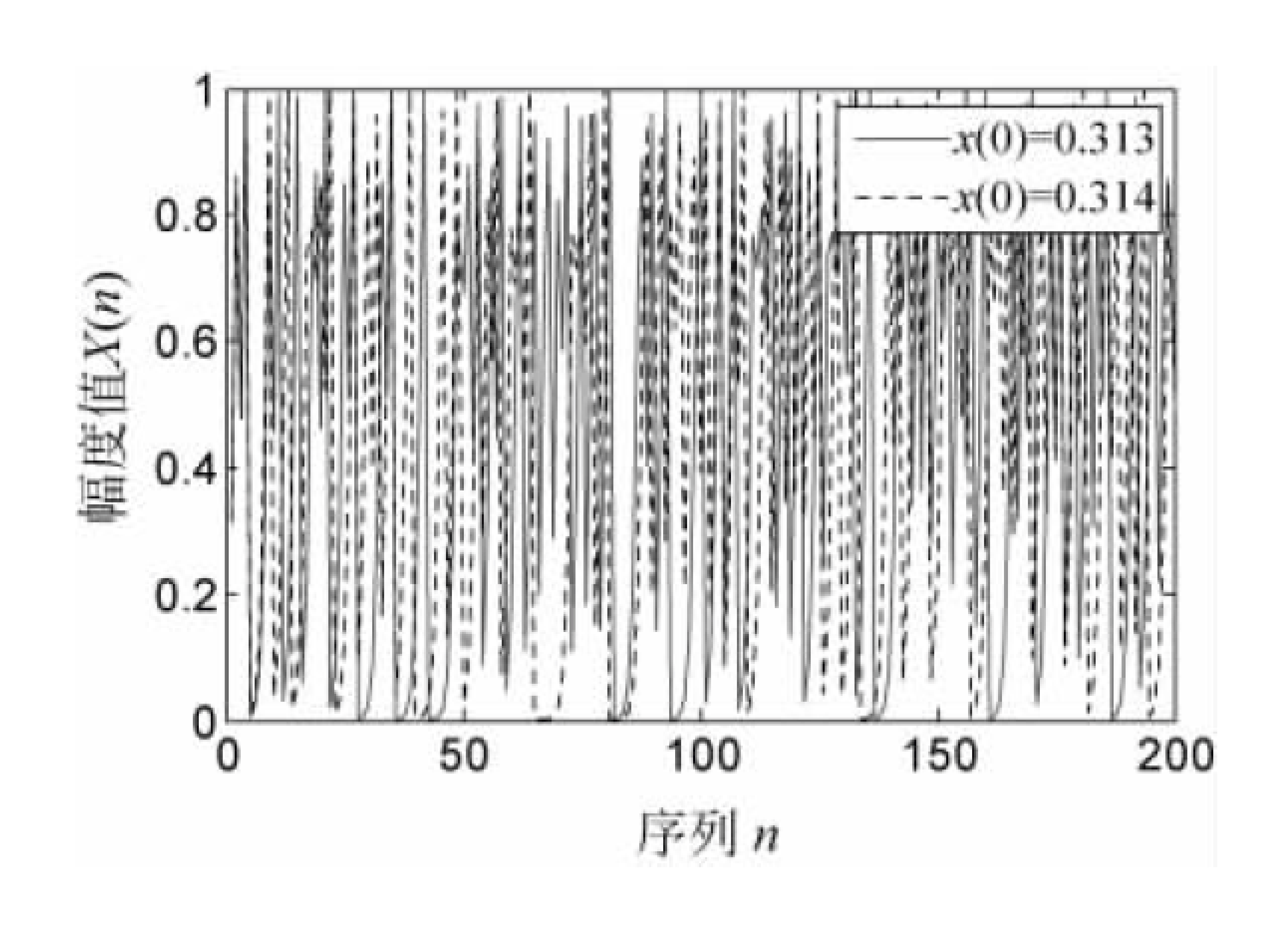



图 3.2　Logistic 映射轨道间指数分离现象

3. 频域的白噪声特性

由于混沌信号的非周期性，混沌信号的频谱表现与随机信号的频谱类似为连续频谱。图 3.3 为 Logistic 映射的频谱图。

图 3.3 表明，除了零频分量外，logistic 混沌映射具有类似白噪声特性的功率谱。

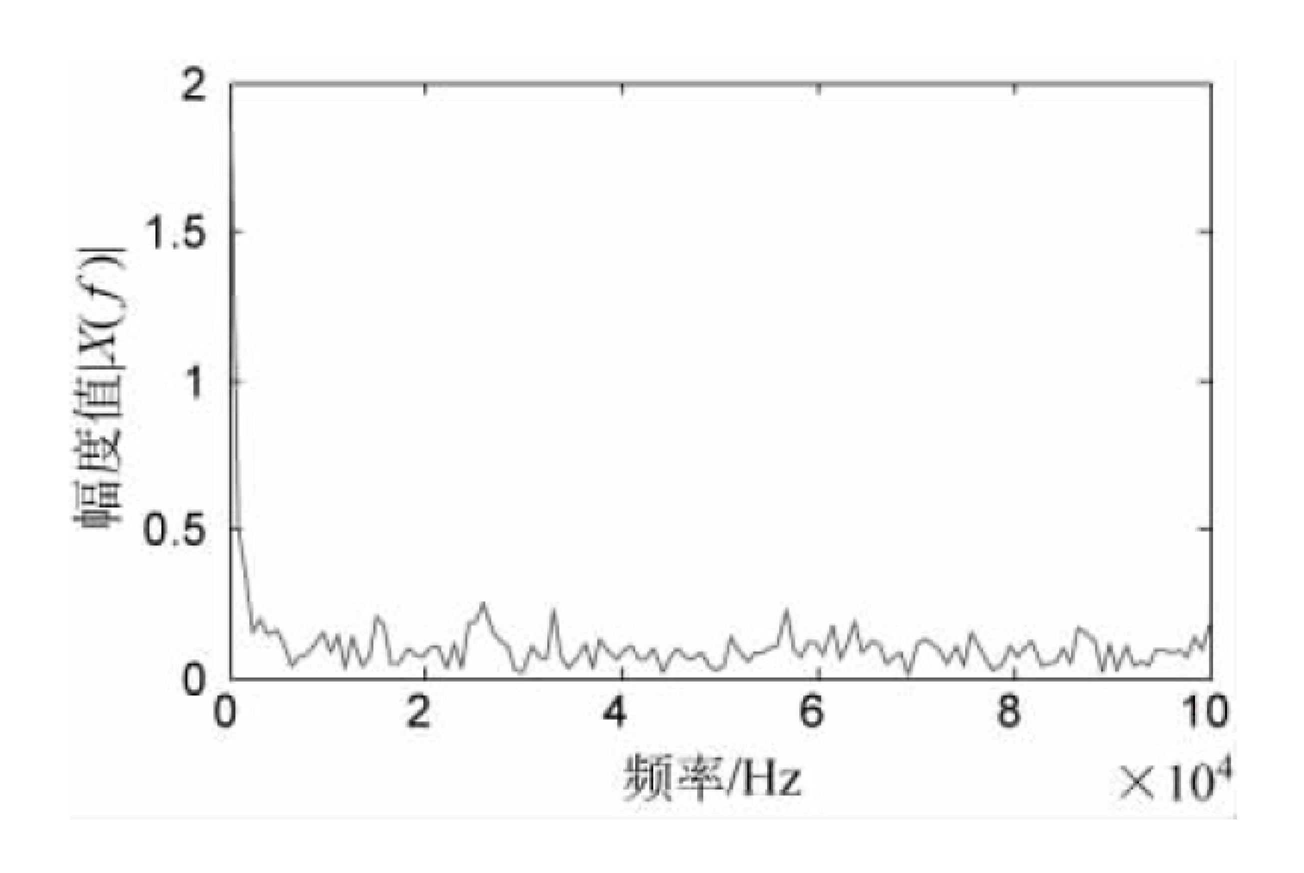



图 3.3　混沌信号的频谱特性

4. 自相关特性

混沌信号的相关特性会随着相关距离的变大而迅速衰减，并表现出与随机信号类似的特性，即具有类似冲激函数的特性，而周期信号的相关函数也是周期的。图 3.4 表明，混沌信号的随机特性与随机信号特性类似，线性统计分析工具很难对混沌信号进行有效分析。

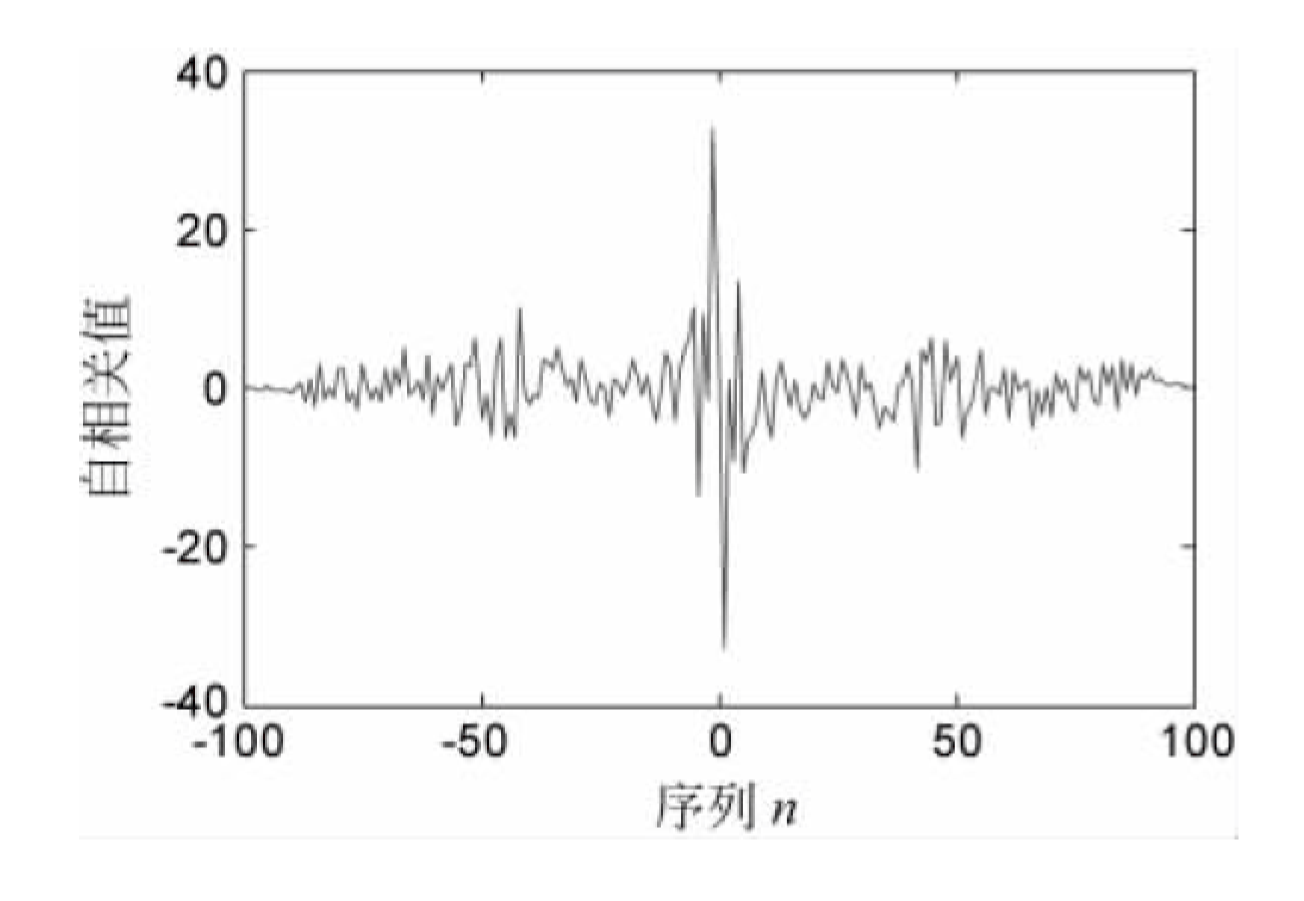



图 3.4　Logistic 混沌序列自相关特性

### 3.1.3　混沌序列的基本特点

1. 混沌映射的特点

离散时间混沌系统的确定性差分方程为

$$s(k+1)=F(s(k),\lambda)\quad k=0,1,2,\cdots \tag{3.1.4}$$

式中，$s(k)\in S$ 是 $n$ 维向量；$S$ 为状态空间；$F$ 为映射函数，将当前的状态 $s(k)$ 映射成下一个状态，以初值 $s(0)$ 开始迭代得到混沌序列 $\{s(k):k=0,1,2,\cdots\}$；$\lambda$ 为外部控制参数。

常用以下几个测度来分析混沌序列的特点，如不动点、Lyapunov 指数、相关函数、功率谱密度等。

（1）不动点

混沌序列映射函数中，如果存在点 $a^*$，满足

$$a^* = F(a^*) \tag{3.1.5}$$

称 $a^*$ 为不动点。如果不动点领域范围，点 $a_0,a_1,a_2,\cdots$，在 $a^*$ 处收敛，则称点 $a^*$ 是局部稳定的，不动点的间距表示为

$$\begin{aligned} D(k+1) &= |a(k+1)-a^*| = |F(a(k))-a^*| \\ &= |\mathrm{d}F(a^*)/\mathrm{d}a|\,D(k) \end{aligned} \tag{3.1.6}$$

不动点局部稳定的条件为

$$\left|\frac{\mathrm{d}}{\mathrm{d}a}F(a^*)\right| < 1 \tag{3.1.7}$$

当该条件满足时，该映射产生的混沌序列是收敛的。

（2）Lyapunov 指数

由于混沌序列对初始值具有很强的敏感性，常用 Lyapunov 指数 $\lambda$ 来描述混沌序列随着初始值的微小变化而呈现出的以指数方式迅速分离的程度。它的定义为[3]：对于相空间中的一点 $a(0)$ 有半径为 $\varepsilon(0)$ 的邻域，该邻域随着动力系统的演化向相空间的各个方向做伸展或收缩，成为一个超椭球，超椭球在各个方向上的轴长为 $\varepsilon_i(t)$，则轨道 $a(t)$ 在第 $i$ 个方向上的 Lyapunov 指数定义为

$$\lambda_i = \lim_{t\to\infty}\left(\lim_{\varepsilon_i(0)\to 0}\frac{1}{t}\ln\frac{\varepsilon_i(t)}{\varepsilon_i(0)}\right) \tag{3.1.8}$$

Lyapunov 指数描述了在局部范围里轨道间的分离程度，当 Lyapunov 指数大于 0 时，轨道间的距离随着时间成指数分离，混沌呈现对初始状态的极度敏感性。

(3) 相关函数

对于一个确定的混沌映射,任意给定的一个初始点 $a(0)$,迭代过后,会得到一个混沌序列$\{a(k): k=0,1,2,\cdots\}$。该混沌序列的均值为

$$E[a]=\lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N-1}a(k) \tag{3.1.9}$$

其自相关函数为

$$R_a(k)=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}[a(n)-E[a]][a(n+k)-E[a]] \tag{3.1.10}$$

自相关函数的归一化形式为

$$r_X(k)=R_X(k)/R_X(0) \tag{3.1.11}$$

对于两个混沌序列$\{a(k): k=0,1,2,\cdots\}$和$\{b(k): k=0,1,2,\cdots\}$,其互相关函数为

$$R_{ab}(k)=\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}[a(n)-E[a]][b(n+k)-E[b]] \tag{3.1.12}$$

式(3.1.12)的归一化为

$$r_{ab}(k)=R_{ab}(k)/\sqrt{R_a(0)R_b(0)} \tag{3.1.13}$$

(4) 功率谱密度

混沌信号具有非周期和能量无限的特点,常用功率谱密度来描述它的统计平均特性。功率谱密度 $G(\omega)$与自相关函数 $R_a(k)$是一对傅里叶变换,即

$$\begin{cases} G(\omega)=\sum_{-\infty}^{+\infty}R_a(k)\mathrm{e}^{-\mathrm{j}\omega k} \\ R_a(k)=\frac{1}{2\pi}\int_{-\pi}^{\pi}G(\omega)\mathrm{e}^{\mathrm{j}\omega k} \end{cases} \tag{3.1.14}$$

两个混沌时间序列的互功率谱密度和互相关函数为

$$\begin{cases} G(\omega)=\sum_{-\infty}^{+\infty}R_{ab}(k)\mathrm{e}^{-\mathrm{j}\omega k} \\ R_{ab}(k)=\frac{1}{2\pi}\int_{-\pi}^{\pi}G(\omega)\mathrm{e}^{\mathrm{j}\omega k} \end{cases} \tag{3.1.15}$$

以上是混沌序列的几个基本特性,表明了混沌序列具有非线性、随机性

和确定性的特点。在实际应用过程中，由于混沌对初始条件极其敏感，而能够得到的初始条件精度有限，造成了混沌运动表现出随机的现象，故很难对实际的混沌序列进行长期预测。但由于混沌运动内部是有规律变换的，随机性现象的本质是确定的，因而可以对混沌序列进行短期的估计和预测。

2. 常见的混沌映射序列[4-6]

除了结构简单的 Logistic 映射外，其他常见的混沌映射如下。

(1) Henon 映射

Henon 映射为二维映射，其映射函数为

$$\begin{cases} x(k+1) = 1 - ax^2(k) + y(k) \\ y(k+1) = bx(k) \end{cases} \tag{3.1.16}$$

式中，$x$，$y$ 为变量；$a$，$b$ 通常为常数，常取 $a=1.4$ 和 $b=0.3$。此时变量 $x$ 处于混沌状态。

(2) Rossler 映射

Rossler 映射被认为是最简单的连续时间混沌系统，它的动力学方程为

$$\begin{cases} \dot{x} = -(z+y) \\ \dot{y} = x + ay \\ \dot{z} = b + (x-c)z \end{cases} \tag{3.1.17}$$

式中，$x$，$y$，$z$ 为变量；$a$，$b$，$c$ 为常数。

(3) Lorenz 映射

Lorenz 映射是描述大气中二维流体对流模型的简化，其三元常微分方程组为

$$\begin{cases} \dot{x} = \sigma(y-x) \\ \dot{y} = rx - xz - y \\ \dot{z} = xy - bz \end{cases} \tag{3.1.18}$$

式中，$x$，$y$，$z$ 为变量；$b$，$\sigma$，$r$ 为常数，通常取 $b=8/3$，$\sigma=10$，$r=28$。

(4) Chebyshev 映射

Chebyshev 映射方程为

$$x(k+1) = \cos(n\cos^{-1}x(k)) \tag{3.1.19}$$

当变量 $x$ 的取值于区间[$-1$,1],该动力系统处于混沌状态。

### 3.1.4 混沌优化算法

由于混沌具有遍历性、对初始条件极其敏感以及内在的规律性等特点,混沌运动能在一定的范围内按其自身的规律不重复地遍历所有状态,故利用混沌变量进行优化搜索可避免陷入局部极小值点,混沌优化算法也成为了一种新颖的优化算法。利用混沌优化算法对混沌变量进行优化通常分两个阶段进行。第一阶段,在优化变量的变化范围内,利用混沌映射依次考察经历的各个点,接收较好的点作为当前的最优点;第二阶段,以当前的最优点为中心,进行很小的扰动,利用其初值敏感性的特点进行细搜索,寻找出全局最优点。

文献[7-9]中用混沌优化算法求解最优值 $\min[J(x)]$,其中 $J(x)$为关于变量 $\boldsymbol{x}$ 的代价函数。在寻优变量 $x$ 的取值范围内,构造混沌变量 $t$ 和寻优变量 $\boldsymbol{x}$ 取值区间的映射关系。通常采用 $x=c+dt$ 的映射形式,其中 $c$、$d$ 是当混沌变量在区间(0,1)遍历时,寻优变量 $\boldsymbol{x}$ 均能在指定范围内变化的常量。

混沌优化算法的迭代步骤可描述如下。

步骤 1:设置控制误差 $e$,给定混沌变量初始值 $t(0)$,计数器 $k=0$。

步骤 2:将 $t(0)$映射到优化变量 $x(0)$的优化区间,$x(0)=c+dt(0)$,并令最优变量 $x_{\text{opt}}=x(0)$,优化函数值 $J_{\text{opt}}=J(0)$。

步骤 3:在优化变量的区间范围内,利用混沌映射函数对优化变量进行搜索得到 $x(k)$和 $J(k)$,如果 $|J(k)-J(k-1)|<e$,则 $x_{\text{opt}}=x(k)$,$J_{\text{opt}}=J(k)$,否则转步骤 4。

步骤 4:令 $k=k+1$,转步骤 3。

步骤 5：若经过上述步骤后 $J_{opt}$ 保持不变，则进行二次载波 $x_{opt}^{(1)}=x_{opt}+\alpha u$，其中 $u$ 为初值设定的一个很小的数，$\alpha$ 是一个可调节的参数。用二次载波后的混沌变量继续迭代，并计算相应的优化函数值 $J_{opt}^{(1)}(k)$，如果 $J_{opt}^{(1)}(k)\leqslant J_{opt}$，则 $J_{opt}=J_{opt}^{(1)}(k)$，$x_{opt}=x_{opt}^{(1)}(k)$，否则，放弃 $x_{opt}^{(1)}(k)$，且 $k=k+1$。

步骤 6：若满足终止条件，则终止混沌迭代，输出最优的混沌变量和最优值，反之则返回步骤 5。

## 3.2　基于混沌优化的正交小波常模盲均衡算法

为了克服 CMA 的收敛速度慢、均方误差较大以及对权向量初始化敏感性的缺陷，本节给出了基于混沌优化的正交小波常数模盲均衡算法(chaos optimization algorithm based WTCMA，CWTCMA)[10]。其原理图如图 3.5 所示。该算法针对 CMA 的局部收敛问题，利用混沌优化算法按混沌运动自身的规律和特性、内在的随机性和遍历性进行高效全局寻优，对正交小波盲均衡器权向量进行优化，使优化权向量位于最优点的邻域范围内，最终使权向量收敛至全局最优解。

### 3.2.1　算法原理

#### 1. 正交小波常数模盲均衡算法

图 3.5 中不含虚线框的部分，就是正交小波常数模盲均衡算法。

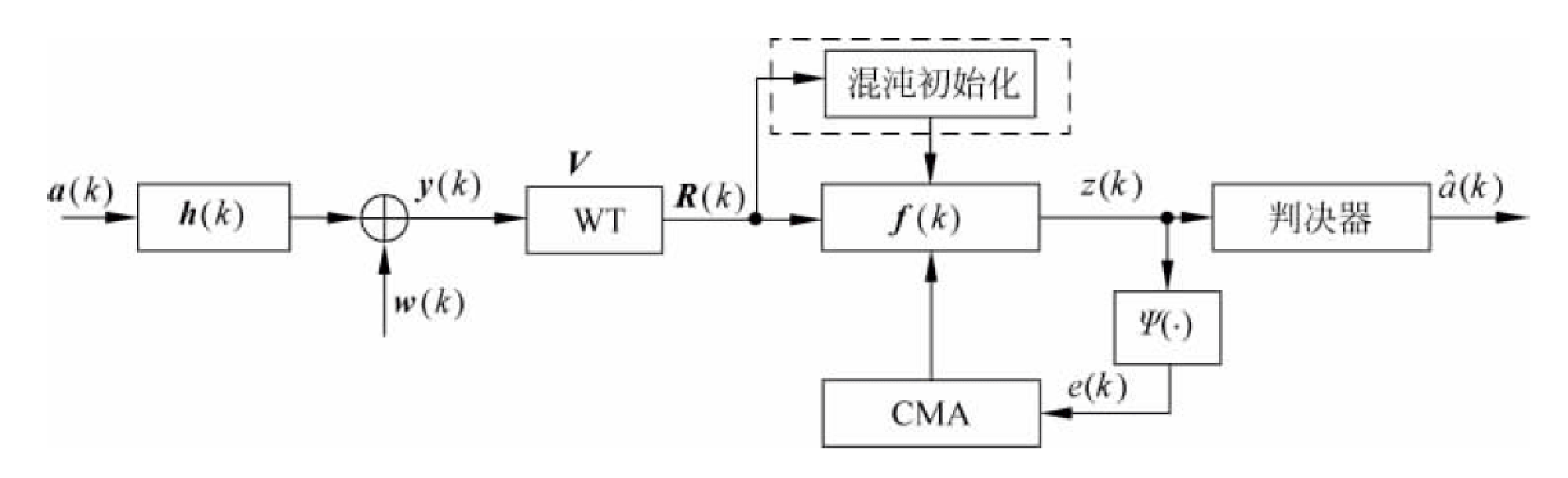



图 3.5　CWTCMA 原理

图 3.5 中，$\boldsymbol{a}(k)$是发射信号；$\boldsymbol{h}(k)$是信道脉冲响应；$\boldsymbol{w}(k)$是加性高斯白噪声；$\boldsymbol{y}(n)$是均衡器接收信号；$\boldsymbol{f}(k)$是均衡器权向量；$\boldsymbol{V}$ 为正交小波变换矩阵；$\boldsymbol{R}(k)$为均衡器的输入信号；$z(k)$是均衡器输出信号；$\hat{a}(k)$是判决装置对 $z(k)$的判决输出信号；$\boldsymbol{\Psi}(\cdot)$是误差生成函数，$e(k)$为误差函数。则有

$$y(k) = \sum_{i=0}^{N_c-1} h_i(k)a(k-i) + w(k) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k) + w(k) \tag{3.2.1}$$

正交小波变换后的输出为

$$\boldsymbol{R}(k) = \boldsymbol{V}\boldsymbol{y}(k) \tag{3.2.2}$$

均衡器输出为

$$z(k) = \boldsymbol{f}^{\mathrm{H}}(k)\boldsymbol{R}(k) \tag{3.2.3}$$

误差函数为

$$e(k) = |z(k)|^2 - R^2 \tag{3.2.4}$$

代价函数为

$$J = E[e^2(k)] = E\{[R^2 - |z(k)|^2]^2\} \tag{3.2.5}$$

通过最速下降法，得到均衡器权向量的迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu\hat{\boldsymbol{R}}^{-1}(k)z(k)[|z(k)|^2 - R^2]\boldsymbol{R}^*(k) \tag{3.2.6}$$

式中，$\mu$ 为步长因子；$\hat{\boldsymbol{R}}^{-1}(k)=\mathrm{diag}[\sigma_{j,0}^2(k),\sigma_{j,1}^2(k),\cdots,\sigma_{j,k_j}^2(k),\sigma_{J+1,0}^2(k),\cdots,\sigma_{J+1,k_j}^2(k)]$为正交小波功率归一化矩阵[11]，其中，diag[]表示对角矩阵，$\sigma_{j,k_j}^2(k)$和 $\sigma_{J+1,k_j}^2(k)$分别表示对小波系数 $u_{j,n}(k)$和尺度系数 $s_{J,n}(k)$的平均功率估计，$u_{j,n}(k)$表示小波空间 $j$ 层分解的第 $n$ 个信号，$s_{J,n}(k)$表示尺度空间中最大分解层数为 $J$ 时第 $n$ 个信号，其递推公式为

$$\begin{cases} \sigma_{j,n}^2(k+1) = \beta\sigma_{j,n}^2(k) + (1-\beta)|u_{j,n}(k)|^2 \\ \sigma_{J+1,n}^2(k+1) = \beta\sigma_{j,n}^2(k) + (1-\beta)|s_{J,n}(k)|^2 \end{cases} \tag{3.2.7}$$

式中，$\beta$ 为平滑因子，且 $0<\beta<1$，一般取略小于 1 的数。称式(3.2.2)～式(3.2.7)为基于正交小波变换的常数模盲均衡算法。

### 2. 权向量的混沌优化

由于混沌运动具有随机性、遍历性、规律性的特点，混沌搜索能在一定的范围内按其自身的规律不重复地遍历每一个状态。混沌优化算法[7-9]就是根据其遍历性和规律性的特点，混沌变量在一定的范围内遍历每一个状态，能够使混沌变量的搜索跳出局部极值点。为避免WTCMA的权向量收敛到局部极小值点，本节将权向量作为优化变量。

在常见的混沌映射函数中，改进的Logistic映射与其他映射相比，具有结构简单、计算量小、使用方便等特点，所以采用改进的Logistic映射作为权向量混沌优化的迭代公式。

该映射的表达式为

$$x(k+1)=1-2x^2(k) \tag{3.2.8}$$

式中，$x$表示混沌变量，映射的相空间范围为$(-1,1)$。

由于WTCMA中权向量的取值范围与改进的Logistic映射的遍历空间不同，式(3.2.9)将第$i$个混沌变量$x_i(k)$映射到相应的第$i$个优化变量中，得到优化变量$f_i(k)$，这样使混沌变量的取值范围“放大”到权向量的取值范围。

$$f_i(k)=c_i+d_i x_i(k) \tag{3.2.9}$$

式中，$c_i$，$d_i$为常数。

为提高权向量的收敛精度，将权向量的实部和虚部分别作为优化变量进行优化，权向量的复数形式可表示为$f_R(k)=f_{iR}(k)+\mathrm{j}f_{iI}(k)$，于是将式(3.2.9)改写为

$$f_{iR}(k)=c_i+d_i x_{iR}(k) \tag{3.2.10a}$$

$$f_{iI}(k)=c_i+d_i x_{iI}(k) \tag{3.2.10b}$$

式中，$x_{iR}(k)$和$x_{iI}(k)$分别为第$i$个权向量的实部和虚部所对应的混沌变量。

对WTCMA权向量进行混沌优化的基本步骤如下。

步骤1：设最大的迭代次数为$M_2$，并对权向量实部和虚部所对应的混

沌变量 $x_{iR}(k)$、$x_{iI}(k)$赋初值 $x_{iR}(0)$和 $x_{iI}(0)$，其中 $i=1,2,\cdots,L$，$L$ 为权向量的长度。

步骤 2：通过式(3.2.10)，将 $x_{iR}(0)$和 $x_{iI}(0)$映射到权向量的优化区间，得到权向量的实部 $f_{iR}(0)$和虚部 $f_{iI}(0)$，令 $f_{iopt}=f_{iR}(0)+\mathrm{j}f_{iI}(0)$，对应的 WTCMA 代价函数 $J_{opt}=J(0)$。

步骤 3：进行混沌搜索，得到 $f_i(k)$和 $J(k)$，如果 $J(k)<J(0)$，则 $f_{iopt}=f_i(k)$，$J_{opt}=J(k)$。

步骤 4：当 $k>M_2$ 时，$f_{iopt}$保持不变，结束；否则，令 $k=k+1$，转到步骤 3。

### 3. 权向量的混合优化

利用混沌优化算法优化权向量理论上可以遍历所有的状态，但优化时间较长，由于 WTCMA 中权向量的迭代利用了最速下降法，而将混沌优化算法与最速下降法有机结合的混合算法具有全局收敛和快速收敛的特点。因此，可以将混沌优化算法与最速下降法结合来优化均衡器权向量。首先通过混沌优化算法全局寻优，使均衡器权向量接近全局最优点，然后采用最速下降法在最优点的邻域范围内局部寻优。利用混合算法搜索，有利于权向量跳出局部最优点，接近全局最优点，并提高收敛的精度。

利用混合算法对盲均衡器权向量 $\boldsymbol{f}$ 进行优化，求解代价函数最小值 $J_{min}$的步骤如下。

步骤 1：设最速下降法和混沌优化的最大迭代次数分别为 $M_1$、$M_2$，混合搜索次数 $M_3$，令计数器 $n=0$，初始权向量 $\boldsymbol{f}(0)$。

步骤 2：以 $\boldsymbol{f}(0)$为初始点，进行 $M_1$ 次最速下降法搜索，得到均衡器优化权向量 $\boldsymbol{f}_{1opt}(k)$和代价函数 $J_{1opt}$。

步骤 3：以 $\boldsymbol{f}_{1opt}(k)$为初始点，进行 $M_2$ 次混沌优化搜索得到 $\boldsymbol{f}_{2opt}$及 $J_{2opt}$。

步骤 4：令 $n=n+1$，如果 $n>M_3$，优化结束；否则，转步骤 5。

步骤 5：如果 $J_{2opt}<J_{1opt}$，则令 $\boldsymbol{f}(k)=\boldsymbol{f}_{2opt}$；如果 $J_{2opt}\geqslant J_{1opt}$，则令 $\boldsymbol{f}(k)=\boldsymbol{f}_{1opt}$，转步骤 1。

基于混沌优化算法和最速下降法的计算步骤,如图 3.6 所示。

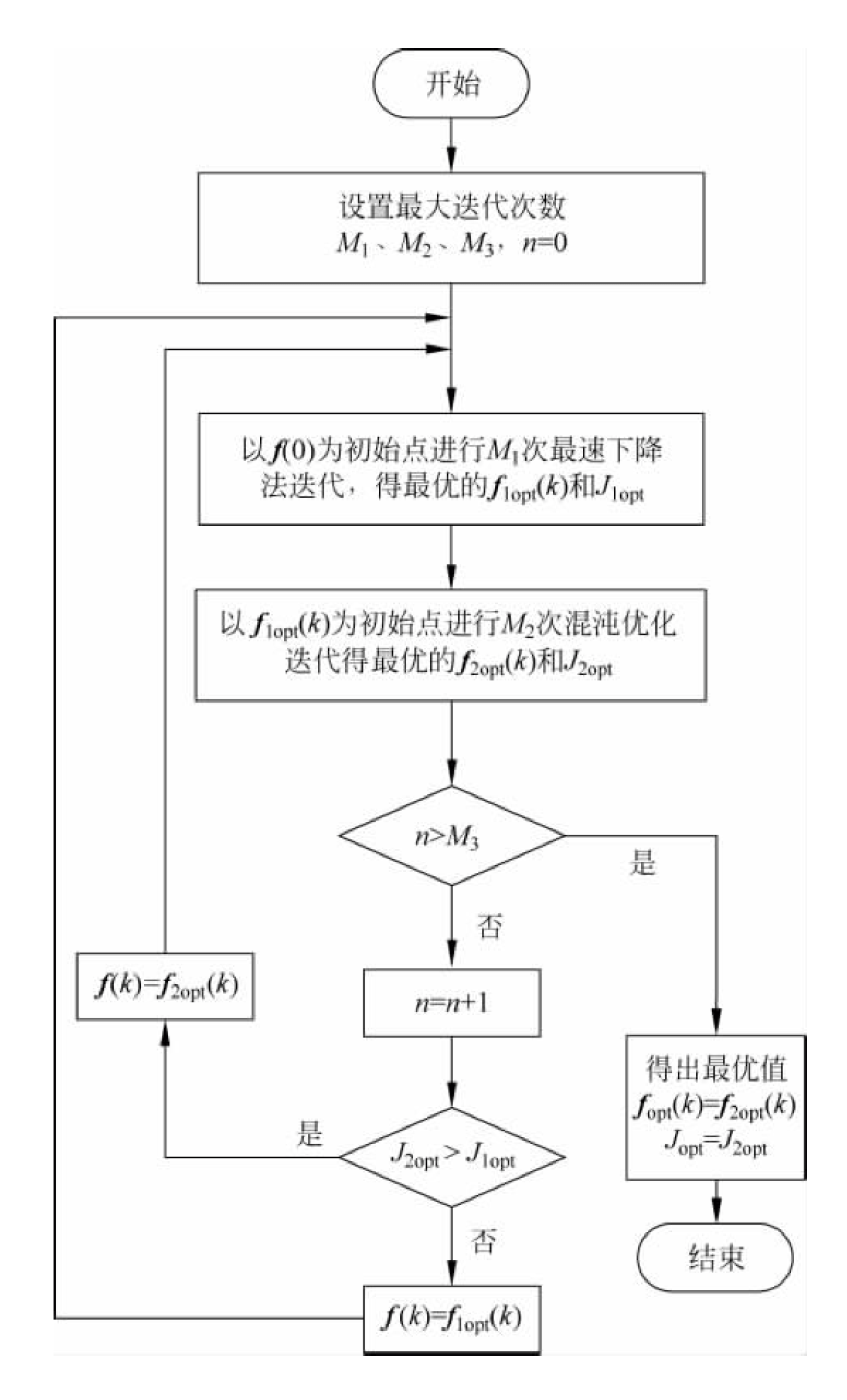



图 3.6　混沌优化算法与最速下降法混合优化

将混沌优化算法与最速下降法结合,利用均衡器接收的一小段数据进行权向量初始化。

令 $\alpha=[0,1,\cdots,m]$,采用 16QAM 时,$m=16$,则 $\eta=16\text{QAM}(\alpha)$,即 $\eta$ 表示对 $\alpha$ 正交幅度调制后的输出信号。根据式(3.2.2),令经过 $M_1$ 次最速下降法迭代优化后的权向量为 $\boldsymbol{f}_{1\text{opt}}$,均衡器权长为 $L$,$k=1,2,\cdots,N$,则

$$\boldsymbol{R}(k) = \boldsymbol{V}\boldsymbol{y}(k+L-1:-1:k) \tag{3.2.11}$$

$$z(k) = \boldsymbol{f}^{\mathrm{H}}(k)\boldsymbol{R}(k) \tag{3.2.12}$$

调制误差为

$$e(k) = \min(|z(k)-\eta|^2) \tag{3.2.13}$$

平均调制误差定义为

$$\mathrm{AME}(n) = \frac{1}{N}\sum_{k=1}^{N}[\min(|z_n(k)-\eta|^2)] \tag{3.2.14}$$

式中,$z_n(k)$表示第 $n$ 次混沌优化时均衡器的输出信号 $z(k)$。利用式(3.2.10)对式(3.2.12)中的权向量 $\boldsymbol{f}$ 的实部和虚部分别进行 $M_2$ 次迭代,每次迭代中调制误差都随权向量的不同而发生改变,除去 $e(k)$中的最大值和最小值,式(3.2.14)修正为

$$\mathrm{AME}(n) = \frac{1}{N-2}\sum_{k=1}^{N}[\min(|z_n(k)-\eta|^2)] \tag{3.2.15}$$

将混沌优化过程中 AME($n$)的最小值赋给 $J_{2\mathrm{opt}}$,其所对应的权向量为 $\boldsymbol{f}_{2\mathrm{opt}}$。算法切换至正交小波常模盲均衡算法的切换条件为

$$J_{2\mathrm{opt}}(n-1) - J_{2\mathrm{opt}}(n) < \zeta \tag{3.2.16}$$

式中,$\zeta$ 为一正数;$J_{2\mathrm{opt}}(n)$表示在第 $n$ 次的混合优化过程中,经过 $M_2$ 次混沌优化后得到的 $J_{2\mathrm{opt}}$,$n$ 的取值范围为 $n=1,2,\cdots,M_3$。在优化过程中,若满足式(3.2.16),则切换到正交小波常模盲均衡算法;若不满足此条件,则经 $M_3$ 次混合优化后,切换到正交小波常模盲均衡算法。

### 3.2.2 算法仿真

为了验证 CWTCMA 的有效性,用水声信道进行仿真研究,并与 CMA、WTCMA 进行比较。仿真实验中,水声信道信道为[0.3132,−0.104,0.8908,0.3134],信噪比为 25dB,均衡器的权长为 16。

**【实验 3.1】** 发射信号为 16QAM,CMA、WTCMA、CWTCMA 中步长 $\mu$ 分别为 0.000 01、0.0002、0.0001,$M_1$、$M_2$、$M_3$ 的值分别为 500、800、20;$N$ 为 20;都采用第 4 个抽头系数为 1,其余的全为 0;$c_i$ 的值都为 0、$d_i$ 的值都

为 1；混沌初始化时采用均衡器输入数据的前 500 点对权向量进行初始化，初始化切换条件 $\zeta$ 为$10^{-5}$。

发射信号为 16PSK，CMA、WT-CMA、CWTCMA 中步长 $\mu$ 分别为 0.001、0.002、0.001，$M_1$、$M_2$、$M_3$ 的值分别为 300、800、20，$N$ 为 20；都采用第 4 个抽头系数为 1，其余的全为 0；$c_i$ 的值都为 0、$d_i$ 的值都为 1；混沌初始化时采用均衡器输入数据的前 300 点对权向量进行初始化，初始化切换条件 $\zeta$ 为$10^{-5}$。

在保证眼图完全清晰睁开的前提下，蒙特卡罗仿真结果如图 3.7 所示。

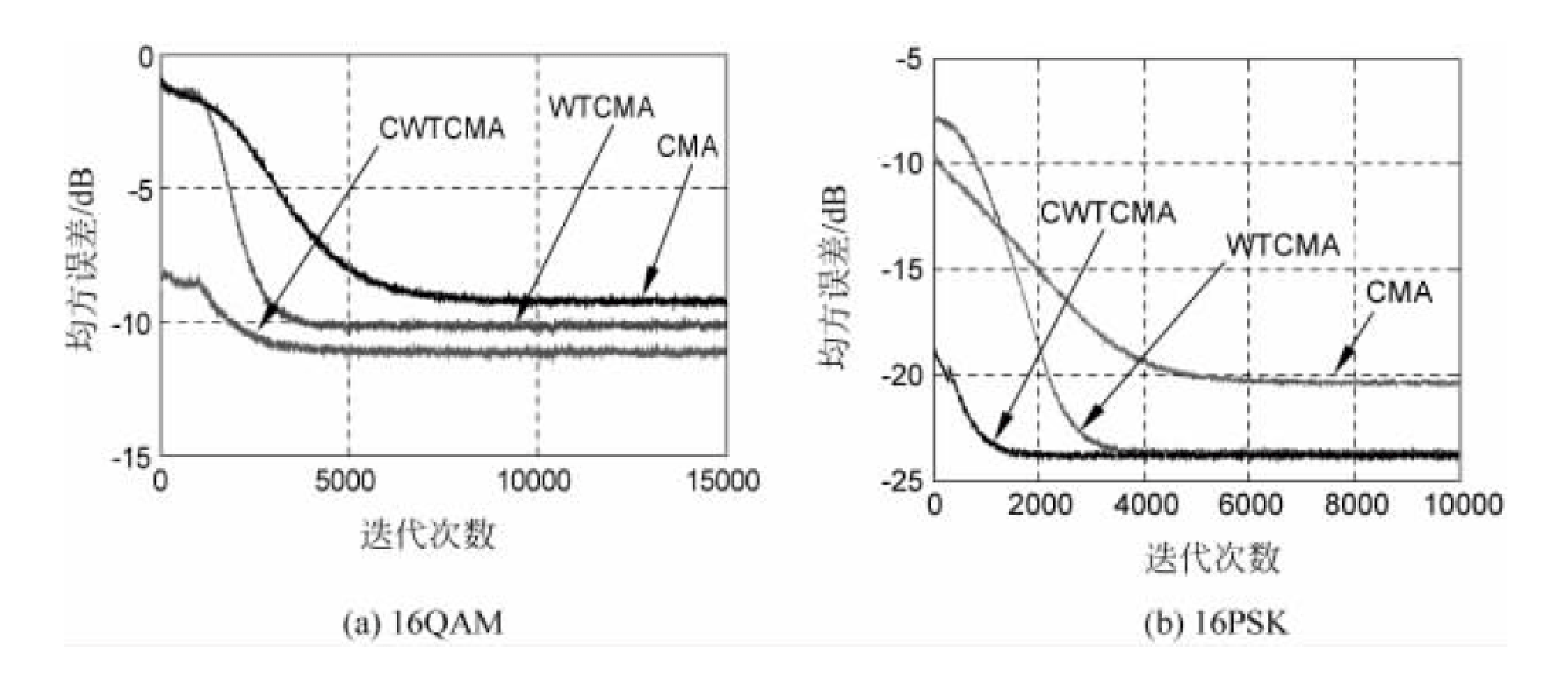



图 3.7　均方误差曲线

图 3.7(a)表明，CWTCMA 收敛后的均方误差比 CMA 约小 2dB，比 WTCMA 约小 0.5dB；CWTCMA 的收敛速度比 CMA 快约 5000 步，比 WTCMA 快约 1000 步。图 3.7(b)表明，在稳态误差上，CWTCMA 比 CMA 减小约 5dB，与 WTCMA 基本相同；在收敛速度上，CWTCMA 比 CMA 快了近 4200 步，比 WTCMA 快了约 1500 步。

## 3.3　基于混沌支持向量机优化的正交小波加权多模盲均衡算法

本节在分析加权多模算法和支持向量机技术的基础上，将正交小波变换引入到加权多模算法中，得到了正交小波加权多模盲均衡算法，并利用支持

向量机对正交小波多模盲均衡算法的权向量进行初始化。为提高支持向量机的学习能力，将支持向量机的参数选取看作参数的组合优化，建立组合优化目标函数，利用混沌优化算法来搜索最优参数值，最终研究了基于混沌支持向量机优化的正交小波加权多模盲均衡算法。

### 3.3.1 加权多模盲均衡算法

在复杂通信中，为提高信道的频带利用率，常对传输的信号进行高阶的QAM调制。对于较高阶QAM信号，采用传统的常数模盲均衡算法进行均衡时，不能匹配高阶的QAM星座图，导致剩余误差较大且收敛速度也很慢，最终均衡性能变差。近年来，针对高阶QAM信号盲均衡算法的研究也取得了很多进展，文献[11-13]从提高收敛速度、降低均衡后的剩余误差等方面做了大量的工作。由于QAM信号的不同调制阶数，星座图结构存在的较大差异，常常需要根据星座图的不同形状设置不同的盲均衡算法[14]，具有一定的局限性。文献[15]提出了一种加权多模盲均衡算法(weighted multi-modulus blind equalization algorithm，WMMA)。该算法引入判决符号的指数幂来调整代价函数中的模值，进一步利用星座图的先验信息，提高了算法的通用性。

当发射高阶QAM信号时，在信道的接收端添加盲均衡器，所构成的简化多模盲均衡系统，如图3.8所示。

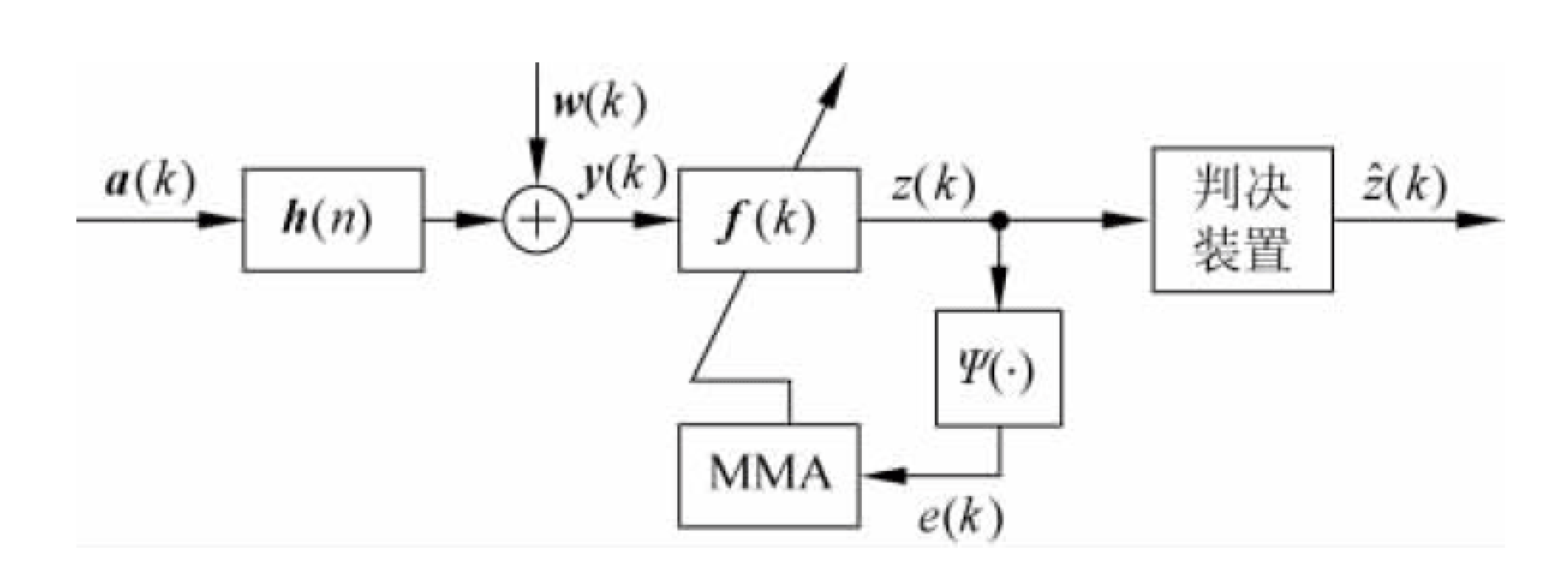



图3.8 多模盲均衡系统框图

图3.8中，$\boldsymbol{a}(k)$为复信源信号且$\boldsymbol{a}(k)=\boldsymbol{a}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{a}_{\mathrm{I}}(k)$，其中，$\boldsymbol{a}_{\mathrm{R}}(k)$和$\boldsymbol{a}_{\mathrm{I}}(k)$分别表示$\boldsymbol{a}(k)$的实部和虚部，j为虚部单位；$\boldsymbol{h}(k)$为信道向量，$\boldsymbol{w}(k)$为高斯白噪声向量，$\boldsymbol{f}(k)$为均衡器权向量；$\boldsymbol{y}(k)$为信道输出信号；$\boldsymbol{\Psi}(\cdot)$为误差生

成函数,$e(k)$为误差函数;$z(k)$为均衡器的输出信号,$\hat{z}(k)$为判决器输出信号。

均衡器输出信号为

$$z(k) = \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k) \tag{3.3.1}$$

文献[10]中多模盲均衡算法(MMA)通过利用均衡器输出信号的幅度信息和相位信息,来消除均衡器接收信号的相位模糊性。将均衡器输出信号 $z(k)$分为实部和虚部,即 $z(k)=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k)$,并在同向和正交方向上获取各自的模值。

MMA 代价函数的形式为

$$J_{\mathrm{MMA}}(\boldsymbol{f}) = E[(z_{\mathrm{R}}^{2}(k) - R_{\mathrm{R}}^{2})^{2} + (z_{\mathrm{I}}^{2}(k) - R_{\mathrm{I}}^{2})^{2}] \tag{3.3.2}$$

式中,$R_{\mathrm{R}}^{2}=E[a_{\mathrm{R}}^{4}(k)/a_{\mathrm{R}}^{2}(k)]$、$R_{\mathrm{I}}^{2}=E[a_{\mathrm{I}}^{4}(k)/a_{\mathrm{I}}^{2}(k)]$分别表示同相方向和正交方向的模值。均衡器权向量的迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu[e_{\mathrm{R,MMA}}(k) + \mathrm{j}e_{\mathrm{I,MMA}}(k)]\boldsymbol{y}^{*}(k) \tag{3.3.3}$$

式中,$e_{\mathrm{R,MMA}}(k)=z_{\mathrm{R}}(k)[z_{\mathrm{R}}(k)-R_{\mathrm{R}}^{2}]$,$e_{\mathrm{I,MMA}}^{2}(k)=z_{\mathrm{I}}(k)[z_{\mathrm{I}}^{2}(k)-R_{\mathrm{I}}^{2}]$,$\mu$ 为迭代步长。式(3.3.1)~式(3.3.3)构成 MMA。MMA 稳态收敛后,式(3.3.3)中误差校正项 $\mu[e_{\mathrm{R,MMA}}(k)+\mathrm{j}e_{\mathrm{I,MMA}}(k)]\boldsymbol{y}^{*}(k)$的均值收敛为零但方差不为零,导致超量均方误差(MSE)比较大,降低了均衡器输出信噪比。一般通过减小 $\mu$ 值和误差项 $e_{\mathrm{R,MMA}}(k)+\mathrm{j}e_{\mathrm{I,MMA}}(k)$来减小稳态误差,而 $\mu$ 值受到收敛速度和精度的制约,不能任意减小,因此常采用降低 MMA 误差项的方法。随着 QAM 阶数的提高,采用 MMA 均衡时,由于 MMA 的误差模型与高阶 QAM 星座图的模型有时不匹配,导致 MSE 随着阶数的提高而越来越大。

文献[15]中的加权多模盲均衡算法(WMMA),充分利用了星座图的先验信息,通过选择合适的误差模型匹配 QAM 星座图模型,达到进一步降低稳态误差的目的。

WMMA 的代价函数形式为

$$\begin{aligned} J(f) = E\{&[z_{\mathrm{R}}^{2}(k) - |\hat{a}_{\mathrm{R}}(k)|^{\lambda_{\mathrm{R}}}R_{\lambda_{\mathrm{R}}}^{2}]^{2} + \\ &[z_{\mathrm{I}}^{2}(k) - |\hat{a}_{\mathrm{I}}(k)|^{\lambda_{\mathrm{I}}}R_{\lambda_{\mathrm{I}}}^{2}]^{2}\} \end{aligned} \tag{3.3.4}$$

式中，$\lambda_R$、$\lambda_I \in [0,2]$为加权因子；$R_{\lambda_R}^2 = E[a_R^4(k)/a_R^2(k)]$，$R_{\lambda_I}^2 = E[a_I^4(k)/a_I^2(k)]$，$\hat{a}_R(k)$、$\hat{a}_I(k)$为判决符号$\hat{a}(k)$的实部和虚部。WMMA 均衡器权向量迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu[e_{R,WMMA}(k) + je_{I,WMMA}(k)]\boldsymbol{y}^*(k) \tag{3.3.5}$$

式中，$e_{R,WMMA}(k) = z_R(k)[z_R^2(k) - |\hat{a}_R(k)|^{\lambda_R} R_{\lambda_R}^2]$，$e_{I,WMMA}(k) = z_I(k)[z_I^2(k) - |\hat{a}_I(k)|^{\lambda_I} R_{\lambda_I}^2]$。

WMMA 与 MMA 的区别在于：WMMA 用模 $R_{R,WMMA}$，$R_{I,WMMA}$ 替代 $R_{MMA}$；$R_{MMA}$是只与信源统计特性相关的常量，而 $R_{R,WMMA}$和 $R_{I,WMMA}$不仅与信源的统计特性有关，还与加权因子 $\lambda$ 和判决器输出的判决符号有关，在权向量迭代过程中自适应地改变取值。

### 3.3.2 支持向量机技术

对于高阶 QAM 调制信号，WMMA 表现出良好的均衡性能，但由于 WMMA 对均衡器权向量初始化很敏感。为避免权向量收敛至局部极小值点，文献[16]提出利用支持向量机(support vector machine，SVM)小样本学习能力强的优势，来求解盲均衡问题。在通过二次规划迭代求解 SVM 盲均衡问题时，计算量随着样本数量变多而变大，因此 SVM 不适合长数据段的均衡。在均衡器的接收端，利用接收的一小段收据来估计均衡器的初始值，而后切换至盲均衡算法，可避免均衡器收敛至局部极小值点，减小了计算复杂度。针对常模信号和多模信号模值不同的特点，本节将利用 SVM 分别对接收为 16PSK 和 16QAM 信号的均衡器权向量进行初始化。

#### 1. PSK 常模信号的 SVM 初始化

对于常模信号，根据 SVM 拟合的结构风险最小化原则[17]，以精度 $\varepsilon$ 估计常模信号的均衡器权向量 $\boldsymbol{f}$，需要最小化的代价函数为

$$J_{\mathrm{CMA}}(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}(k)\|^2 + C\sum_{k=1}^{N}|R^2 - [\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2|_{\varepsilon} \tag{3.3.6}$$

式中，$C$ 为惩罚系数。

根据 Vapnik 的 $\varepsilon$ 不敏感损失函数，有

$$|R^2 - [\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2|_{\varepsilon} = \max\{0, |[R^2 - \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2| - \varepsilon\} \tag{3.3.7}$$

引入松弛变量 $\xi(k)$和 $\tilde{\xi}(k)$，式(3.3.6)的最小值可以通过求解以下约束最优化问题得到：

最小化

$$L[\boldsymbol{f},\xi(k),\tilde{\xi}(k)] = \frac{1}{2}\|\boldsymbol{f}(k)\|^2 + C\sum_{k=1}^{N}[\xi(k) + \tilde{\xi}(k)] \tag{3.3.8}$$

约束条件为

$$\begin{aligned} &[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2 - R^2 \leqslant \varepsilon + \xi(k) \\ &R^2 - [\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2 \leqslant \varepsilon + \tilde{\xi}(k) \\ &\xi(k),\tilde{\xi}(k) \geqslant 0 \end{aligned} \tag{3.3.9}$$

令均衡器输出 $z(k)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)$是固定的，可以将式(3.3.8)中的二次约束条件式(3.3.9)改为线性约束，即

$$\begin{aligned} &z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] - R^2 \leqslant \varepsilon + \xi(k) \\ &R^2 - z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] \leqslant \varepsilon + \tilde{\xi}(k) \end{aligned} \tag{3.3.10}$$

于是，式(3.3.8)的最优问题可以转化为：给定 $C$ 和$\varepsilon$，求以下拉格朗日鞍点问题[18]。

$$\begin{aligned} L(\boldsymbol{f},\xi,\tilde{\xi},\alpha,\tilde{\alpha},\gamma,\tilde{\gamma}) = &\frac{1}{2}\|\boldsymbol{f}(k)\|^2 + C\sum_{k=1}^{N}[\xi(k) + \tilde{\xi}(k)] - \\ &\sum_{k=1}^{N}[\gamma(k)\xi(k) + \tilde{\gamma}(k)\tilde{\xi}(k)] - \\ &\sum_{k=1}^{N}\alpha(k)\{R^2 - z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] + \varepsilon + \xi(k)\} - \\ &\sum_{k=1}^{N}\tilde{\alpha}(k)\{z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] - R^2 + \varepsilon + \tilde{\xi}(k)\} \end{aligned} \tag{3.3.11}$$

式中，$\alpha(k)\geqslant 0$，$\tilde{\alpha}(k)\geqslant 0$，$\gamma(k)\geqslant 0$，$\tilde{\gamma}(k)\geqslant 0$。

$$\frac{\partial L}{\partial \boldsymbol{f}}=0\Rightarrow \boldsymbol{f}_{\text{opt}}=\sum_{k=1}^{N}[\tilde{\alpha}(k)-\alpha(k)]z(k)\boldsymbol{y}(k) \tag{3.3.12}$$

$$\frac{\partial L}{\partial \xi(k)}=0\Rightarrow C-\gamma(k)-\alpha(k)=0 \quad \forall k=1,2,\cdots,N \tag{3.3.13}$$

$$\frac{\partial L}{\partial \tilde{\xi}(k)}=0\Rightarrow C-\tilde{\gamma}(k)-\tilde{\alpha}(k)=0 \quad \forall k=1,2,\cdots,N \tag{3.3.14}$$

$$\frac{\partial L}{\partial \varepsilon}=0\Rightarrow \alpha(k)+\tilde{\alpha}(k)=0 \quad \forall k=1,2,\cdots,N \tag{3.3.15}$$

将式(3.3.12)代入$\frac{1}{2}\|\boldsymbol{f}(k)\|^2$中，得

$$\begin{aligned}\frac{1}{2}\|\boldsymbol{f}(k)\|^2&=\frac{1}{2}\boldsymbol{f}(k)\boldsymbol{f}^*(k)\\&=\frac{1}{2}\sum_{i=1}^{N}[\tilde{\alpha}(i)-\alpha(i)]z(i)y(i)\cdot\sum_{j=1}^{N}\{[\tilde{\alpha}(j)-\alpha(j)][z(j)y(j)]\}^*\\&=\frac{1}{2}\sum_{i,j=1}^{N}[\tilde{\alpha}(i)-\alpha(i)][\tilde{\alpha}(j)-\alpha(j)][z(i)y(i)][z(j)y(j)]^*\end{aligned} \tag{3.3.16}$$

由式(3.3.13)和式(3.3.14)，得

$$\gamma(k)=C-\alpha(k) \tag{3.3.17}$$

$$\tilde{\gamma}(k)=C-\tilde{\alpha}(k) \tag{3.3.18}$$

将式(3.3.16)、式(3.3.17)和式(3.3.18)代入到式(3.3.11)，得式(3.3.11)的最小化问题可表示为

$$\begin{aligned}W(\alpha,\tilde{\alpha})=&\varepsilon\sum_{i=1}^{N}[\alpha(i)+\tilde{\alpha}(i)]-R^2\sum_{i=1}^{N}[\tilde{\alpha}(i)-\alpha(i)]+\\&\frac{1}{2}\sum_{i,j=1}^{N}[\tilde{\alpha}(i)-\alpha(i)][\tilde{\alpha}(j)-\alpha(j)][z(i)y(i)][z(j)y(j)]^*\end{aligned} \tag{3.3.19}$$

权向量更新公式为

$$\boldsymbol{f}(k+1)=\lambda\boldsymbol{f}(k)+(1-\lambda)\boldsymbol{f}_{\text{opt}} \tag{3.3.20}$$

式中，$\lambda$ 为接近于 1 的常数。在 SVM 学习的过程中，利用式(3.3.20)调节

权向量 $\boldsymbol{f}(k)$，使输出信号 $z(k)$ 的模值接近于 $R^2$。

在常模信号中，利用 SVM 初始化后切换到其他算法的条件为

$$\frac{1}{N}\sum_{k=1}^{N}[R^2 - |z(k)|^2] < T \tag{3.3.21}$$

式中，$T$ 为一个取值很小的正数。

2. QAM 多模信号的 SVM 初始化

对于 QAM 信号，假定均衡器输入为 16QAM 信号。

令 $\alpha=[1,2,\cdots,m]$，采用 16QAM 时，$m=16$，则 $\eta=16\text{QAM}(\alpha)$，即 $\eta$ 表示对 $\alpha$ 正交幅度调制后的输出信号，令 $\eta=[\eta_1,\eta_2,\cdots,\eta_m]$，$\eta_m$ 为对应第 $m$ 个输入信号的调制输出信号。令均衡器的输入序列为 $y(k)$，$k=1,2,\cdots,N$，也就是利用 $N$ 个均衡器输入信号，通过 SVM 对均衡器权向量进行初始化。

假定 $\boldsymbol{R}^{2'}=[R^{2'}(1),R^{2'}(2),\cdots,R^{2'}(k),\cdots,R^{2'}(N)]$，$k=1,2,\cdots,N$。均衡器的第 $k$ 个输出信号为 $z(k)$，有

$$e_i(k) = |z(k) - \eta_i|^2 \tag{3.3.22}$$

式中，$\eta_i$ 为 $\eta$ 中的第 $i$ 个元素，$i=1,2,\cdots,16$，取最小的 $e_i(k)$ 所对应的 $\eta_i^2$ 为 $R'(k)$ 的值，也就是均衡器的输出信号 $z(k)$ 在 QAM 星座图中，离 $\eta_i$ 点的距离最近。

对于一个高阶 QAM 信号，根据 SVM 拟合结构风险最小化原则，以 $\varepsilon$ 估计权向量 $\boldsymbol{f}$，最小化代价函数为

$$J_{\text{QAM}}(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}(k)\|^2 + C\sum_{k=1}^{N}|R^{2'}(k) - [\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2|_{\varepsilon} \tag{3.3.23}$$

根据 Vapnik 的 $\varepsilon$ 不敏感损失函数，有

$$|R^{2'}(k) - [\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2|_{\varepsilon} = \max\{0, |[R^{2'}(k) - \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]^2| - \varepsilon\} \tag{3.3.24}$$

线性约束为

$$\begin{aligned} z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] - R^{2'}(k) &\leqslant \varepsilon + \xi(k) \\ R^{2'}(k) - z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] &\leqslant \varepsilon + \tilde{\xi}(k) \end{aligned} \tag{3.3.25}$$

最优问题可以转化为：给定 $C$ 和 $\varepsilon$，求以下拉格朗日鞍点问题。

$$
\begin{aligned}
L(\boldsymbol{f},\xi,\tilde{\xi},\alpha,\tilde{\alpha},\gamma,\tilde{\gamma}) = & \frac{1}{2}\|\boldsymbol{f}\|^2 + C\sum_{n=1}^{N}[\xi(k)+\tilde{\xi}(k)] - \\
& \sum_{n=1}^{N}[\gamma(k)\xi(k)+\tilde{\gamma}(k)\tilde{\xi}(k)] - \\
& \sum_{k=1}^{N}\alpha(k)\{R^{2'}(k) - z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)]+\varepsilon+\xi(k)\} - \\
& \sum_{k=1}^{N}\tilde{\alpha}(k)\{z(k)[\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)] - R^{2'}(k)+\varepsilon+\tilde{\xi}(k)\}
\end{aligned}
\tag{3.3.26}
$$

与常模算法推导过程相似，式(3.3.26)最小化可表示为

$$
\begin{aligned}
W(\alpha,\tilde{\alpha}) = & \varepsilon\sum_{i=1}^{N}[\alpha(i)+\tilde{\alpha}(i)] - \sum_{i=1}^{N}R^{2'}(i)[\tilde{\alpha}(i)-\alpha(i)] + \\
& \frac{1}{2}\sum_{i,j=1}^{N}[\tilde{\alpha}(i)-\alpha(i)][\tilde{\alpha}(j)-\alpha(j)][z(i)y(i)][z(j)y(j)]^{*}
\end{aligned}
\tag{3.3.27}
$$

通过式

$$
\boldsymbol{f}(k+1) = \lambda\boldsymbol{f}(k) + (1-\lambda)\boldsymbol{f}_{\mathrm{opt}} \tag{3.3.28}
$$

不断的调节权向量 $\boldsymbol{f}$，使 $z(k)$ 不断接近其在 QAM 星座图中的点 $\eta_i$。

而利用支持向量机切换到其他算法的条件为

$$
\frac{1}{N}\sum_{k=1}^{N}[R^{2'}(k) - |y(k)|^2] < T \tag{3.3.29}
$$

### 3.3.3 正交小波加权多模盲均衡算法

对于高阶 QAM 调制方式，MMA 利用了均衡器输出信号的幅度和相位信息，提高了稳态收敛性能，但 MMA 中权向量的同相和正交分量都是利用单一的判决圆进行调整的，QAM 阶数越高，信道均衡性能越差，收敛速度和均方误差达不到理想的效果。WMMA 引入判决符号的指数幂来调整代价函数中的模值，进一步利用了高阶 QAM 星座图的信息，对信道具有

很好的均衡性能。为加快 WMMA 的收敛速度，利用正交小波变换的去相关性降低均衡器输入信号的自相关性，得到了基于正交小波变换的加权多模盲均衡算法(wavelet transform based WMMA，WTWMMA)。该算法原理如图 3.9 所示。

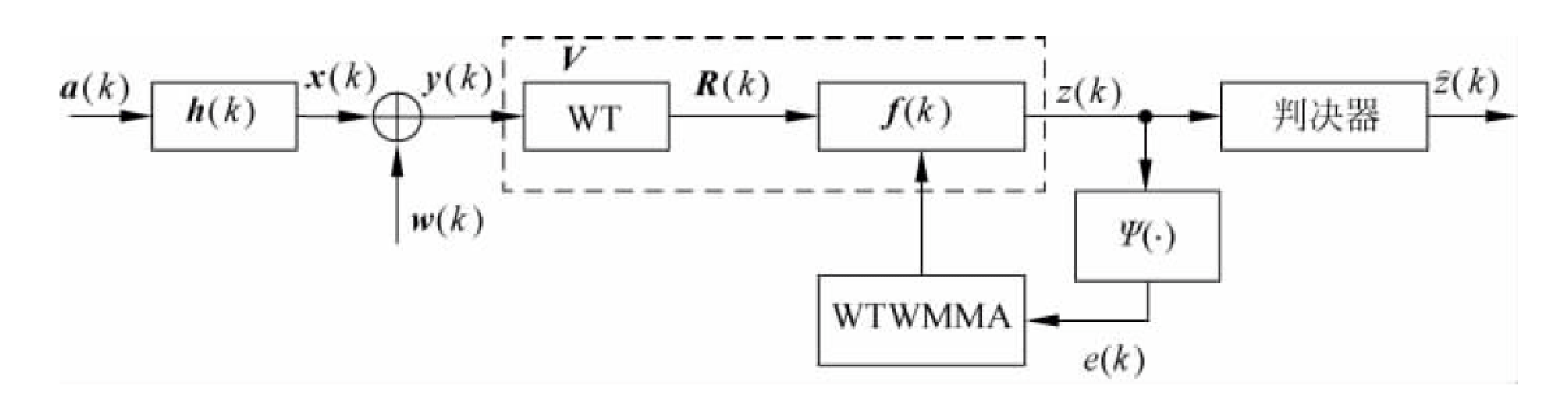



图 3.9　WTWMMA 原理

图 3.9 中，$\boldsymbol{a}(k)$为复信源信号，可表示为 $\boldsymbol{a}(k)=\boldsymbol{a}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{a}_{\mathrm{I}}(k)$，且 $\boldsymbol{a}_{\mathrm{R}}(k)$和 $\boldsymbol{a}_{\mathrm{I}}(k)$分别表示 $\boldsymbol{a}(k)$的实部和虚部，j 为虚部单位；$\boldsymbol{h}(k)$为信道向量；$\boldsymbol{w}(k)$为高斯白噪声向量；$\boldsymbol{\Psi}(\cdot)$为误差生成函数，$e(k)$为误差函数；$\boldsymbol{f}(k)$为均衡器的权向量；$\boldsymbol{y}(k)$为信道输出信号；$\boldsymbol{V}$ 为正交小波变换矩阵；$\boldsymbol{R}(k)$为均衡器的输入信号；$z(k)$为均衡器的输出信号，$\hat{z}(k)$为判决器输出信号。

均衡器输入为

$$\boldsymbol{R}(k)=\boldsymbol{V}\boldsymbol{y}(k) \tag{3.3.30}$$

均衡器输出为

$$z(k)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{R}(k) \tag{3.3.31}$$

设 $z_{\mathrm{R}}(k)$和 $z_{\mathrm{I}}(k)$分别为均衡器输出信号 $z(k)$的实部和虚部，则

$$z(k)=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k) \tag{3.3.32}$$

代价函数为

$$J_{\mathrm{WTWMMA}}=E\{[z_{\mathrm{R}}^{2}(k)-|\hat{z}_{\mathrm{R}}(k)|^{\lambda_{\mathrm{R}}}R_{\lambda_{\mathrm{R}}}^{2}]^{2}+[z_{\mathrm{I}}^{2}(k)-|\hat{z}_{\mathrm{I}}(k)|^{\lambda_{\mathrm{I}}}R_{\lambda_{\mathrm{I}}}^{2}]^{2}\} \tag{3.3.33}$$

在 WTWMMA 中，均衡器的输出误差为

$$\begin{cases} e_{\mathrm{R,WTWMMA}}(k)=z_{\mathrm{R}}(k)(z_{\mathrm{R}}^{2}(k)-|\hat{z}_{\mathrm{R}}(k)|^{\lambda_{\mathrm{R}}}R_{\lambda_{\mathrm{R}}}^{2}) \\ e_{\mathrm{I,WTWMMA}}(k)=z_{\mathrm{I}}(k)(z_{\mathrm{I}}^{2}(k)-|\hat{z}_{\mathrm{I}}(k)|^{\lambda_{\mathrm{I}}}R_{\lambda_{\mathrm{I}}}^{2}) \end{cases} \tag{3.3.34}$$

式中，$e_{\mathrm{R,WTWMMA}}(k)$、$e_{\mathrm{I,WTWMMA}}(k)$分别为均衡器输出误差 $e_{\mathrm{WTWMMA}}(k)$的实部和虚部；$R^2_{\lambda_{\mathrm{R}},\mathrm{WMMA}}=E(a^4_{\mathrm{R}}(k))/E(|a_{\mathrm{R}}(k)|^{2+\lambda_{\mathrm{R}}})$，$R^2_{\lambda_{\mathrm{I}},\mathrm{WMMA}}=E(a^4_{\mathrm{I}}(k))/E(|a_{\mathrm{I}}(k)|^{2+\lambda_{\mathrm{I}}})$。

WTWMMA 均衡器权向量的迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu\hat{\boldsymbol{R}}^{-1}(k)[e_{\mathrm{R,WTWMMA}}+\mathrm{j}e_{\mathrm{I,WTWMMA}}(k)]\boldsymbol{R}^*(k) \tag{3.3.35}$$

式中，$\mu$ 为步长；$\hat{\boldsymbol{R}}^{-1}(k)=\mathrm{diag}[\sigma^2_{1,0}(k),\sigma^2_{1,1}(k),\cdots,\sigma^2_{I,k_I-1}(k),\sigma^2_{I+1,0}(k),\cdots,\sigma^2_{I+1,k_I-1}(k)]$，且 $\sigma^2_{i,n}(k)$，$\sigma^2_{I+1,n}(k)$分别表示对小波变换系数 $u_{i,n}(k)$、尺度变换系数 $s_{I,n}(k)$的平均功率估计，其递推估计公式为

$$\hat{\sigma}^2_{i,n}(k+1)=\beta\hat{\sigma}^2_{i,n}(k)+(1-\beta)\,|\,u_{i,n}(k)\,|^2 \tag{3.3.36}$$

$$\hat{\sigma}^2_{I+1,n}(k+1)=\beta\hat{\sigma}^2_{I+1,n}(k)+(1-\beta)\,|\,s_{I,n}(k)\,|^2 \tag{3.3.37}$$

式中，$\beta$ 是平滑因子，且 $0<\beta<1$。式(3.3.30)～式(3.3.37)称为正交小波变换加权多模盲均衡算法。

### 3.3.4 基于混沌支持向量机优化的正交小波加权多模盲均衡算法

由于 WTWMMA 权向量采用最速下降法进行迭代，与 CMA 类似，容易陷入局部极小值点。利用支持向量机对 WTWMMA 权向量进行初始化，可以提高收敛速度并避免陷入局部极小值点。在支持向量机初始化权向量的过程中，将支持向量机参数的选取[19-22]看作参数的组合优化，建立组合优化目标函数；采用混沌优化算法来搜索最优的目标函数值，以提高支持向量机的拟合能力。本节将混沌优化算法、支持向量机和正交小波变换引入到 WMMA 中，从而获得了基于混沌支持向量机优化的正交小波加权多模盲均衡算法(chaos and support vector machines optimization based wavelet transform weighted multi-modulus blind equalization algorithm，CSVM-WTWMMA)[23]，如图 3.10 所示。

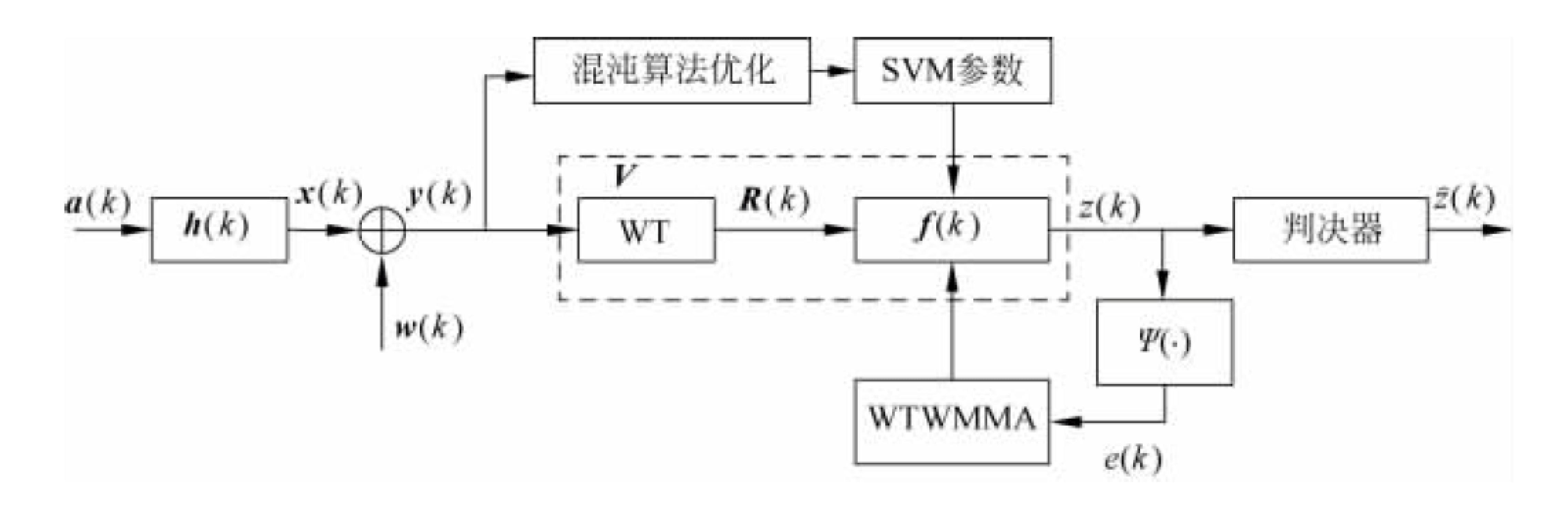



图 3.10 CSVM-WTWMMA 原理

图 3.10 表明,优化可以分为两个方面：其一,由支持向量机优化均衡器初始权向量；其二,混沌算法优化支持向量机参数。

1. 混沌算法优化支持向量机参数

由于 WTWMMA 中的权向量迭代利用最速下降法,易陷入局部极小值点,利用支持向量机将盲均衡问题转化为全局最优支持向量机回归问题。根据多模信号模值特点,利用 SVM 对均衡器接收的一小段信号,对均衡器权向量进行初始化。

令信源的发射信号为 $a(k)$,则均衡器的接收信号为

$$y(k) = \sum_i h(i)a(k-i) + w(k) \tag{3.3.38}$$

在初始化过程中,均衡器第 $k$ 个输出信号为 $z(k)$,有

$$e_i(k) = |z(k) - \eta_i|^2 \tag{3.3.39}$$

假定 $\boldsymbol{R}^{2'}=[R^{2'}(1),R^{2'}(2),\cdots,R^{2'}(k),\cdots,R^{2'}(N)]$,$k=1,2,\cdots,N$,则取式(3.3.39)中最小的 $e_i(k)$所对应的 $\eta_i^2$ 为 $R^{2'}(k)$的值。也就是说,当均衡器输出信号 $z(k)$在 QAM 星座图中时,离 $\eta_i$ 点的距离最近。

对于高阶 QAM 信号,利用支持向量机初始化权向量,以精度 $\varepsilon$ 估计均衡器权向量 $\boldsymbol{f}$,需最小化的代价函数为

$$J_{\mathrm{WTWMMA}}(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}\|^2 + C\sum_{k=1}^{N} |R^{2'}(k) - [\boldsymbol{f}^{\mathrm{T}}\boldsymbol{y}(k)]^2|_{\varepsilon} \tag{3.3.40}$$

式中,$C>0$ 是惩罚变量,根据文献[18],惩罚变量 $C$ 为

$$C = E[g(k)] + 3\sigma_g \tag{3.3.41}$$

式中，$g(k)=|y(k)|^2$。

参数 $\varepsilon$ 的值可由下式确定

$$\varepsilon = 3\sqrt{\sigma_n^2 \frac{\ln N}{N}} \tag{3.3.42}$$

式中，$\sigma_n^2$ 为噪声方差。

根据式(3.3.41)和式(3.3.42)，求以下拉格朗日鞍点，即

$$\begin{aligned} L(\boldsymbol{f},\xi,\tilde{\xi},\alpha,\tilde{\alpha},\gamma,\tilde{\gamma}) = & \frac{1}{2}\|\boldsymbol{f}\|^2 + C\sum_{k=1}^{N}[\xi(k)+\tilde{\xi}(k)] - \\ & \sum_{k=1}^{N}[\gamma(k)\xi(k)+\tilde{\gamma}(k)\tilde{\xi}(k)] - \\ & \sum_{k=1}^{N}\alpha(k)\{R^{2'}(k) - z(k)[\boldsymbol{f}^{\mathrm{T}}\boldsymbol{y}(k)] + \varepsilon + \xi(k)\} - \\ & \sum_{k=1}^{N}\tilde{\alpha}(k)\{z(k)[\boldsymbol{f}^{\mathrm{T}}\boldsymbol{y}(k)] - R^{2'}(k) + \varepsilon + \tilde{\xi}(k)\} \end{aligned} \tag{3.3.43}$$

对式(3.3.43)进行鞍点求解，即最小化求解 $\boldsymbol{f}$，$\xi(k)$和 $\tilde{\xi}(k)$，最大化求解 $\alpha(k)$，$\tilde{\alpha}(k)$，$\gamma(k)$和 $\tilde{\gamma}(k)$，得

$$\boldsymbol{f}_{\mathrm{opt}} = \sum_{k=1}^{N}[\tilde{\alpha}(k)-\alpha(k)]z(k)\boldsymbol{y}(k) \tag{3.3.44}$$

均衡器最优权向量按式(3.3.44)确定，在支持向量机学习过程中，权向量更新公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + (1-\lambda)\boldsymbol{f}_{\mathrm{opt}} \tag{3.3.45}$$

式中，$\lambda$ 为接近于 1 的常数，$k$ 表示支持向量机的学习次数。

初始化过程中，平均调制误差定义为

$$\mathrm{AME}(n) = \frac{1}{N}\sum_{k=1}^{N}[R'_n(k) - |z_n(k)|^2] \tag{3.3.46}$$

式中，$z_n(k)$表示支持向量机在第 $n$ 次学习过程中，均衡器输出信号。初始化过程切换到正交小波加权多模盲均衡算法的条件为

$$\mathrm{AME}(n-1) - \mathrm{AME}(n) < \zeta \tag{3.3.47}$$

式中，$\zeta$ 是一个取值很小的正数。

2. 混沌算法优化 SVM 参数

混沌运动具有遍历性、随机性和规律性的特点，混沌搜索能在一定的范围内按其自身的规律不重复地遍历每一个状态。混沌优化算法根据其遍历性和规律性的特点，采用混沌变量在一定的范围内遍历，最终搜索到目标函数的最优值。利用 SVM 初始化均衡器权向量时，SVM 参数的取值决定了其学习能力和泛化能力[24-26]，将 SVM 参数的选取看作参数的组合优化，建立组合优化目标函数，采用混沌优化算法来搜索最优的目标函数值。

现将 SVM 的参数 $C$ 和 $\varepsilon$ 选取看作参数的组合优化，将 AME($n$)作为组合优化的目标函数，利用混沌优化算法来搜索最优的目标函数值，从而找到合适的参数取值。将混沌优化算法优化参数后的 SVM 为混沌支持向量机(chaos& support vector machines，CSVM)。

由 Logistic 映射的混沌变量迭代公式为

$$x(k+1)=\mu x(k)[1-x(k)] \tag{3.3.48}$$

式中，$x(k)$为混沌变量；$\mu$ 为一常数，当 $\mu=4$ 时系统完全处于混沌状态。以 AME($n$)最小值作为 SVM 回归与参考模型之间的偏差，即

$$\begin{gathered}\min f(z_1,z_2)=\min(\text{AME})\\ a_i\leqslant z_i\leqslant b_i,\quad i=1,2\end{gathered} \tag{3.3.49}$$

式中，$z_1$，$z_2$ 为优化变量，分别对应于支持向量机参数 $C$ 和 $\varepsilon$；$[a_i,b_i]$为变量 $z_i$ 的定义域。基于混沌优化算法选取 SVM 参数的步骤如下。

步骤 1：初始化变量。令混沌搜索次数为 $M_1$，混沌再搜索次数 $M_2$；计数器 $n_1=0$，$n_2=0$；给优化的混沌变量 $x_i$ 赋初值 $x_i=x_i(0)$，$x_{i\text{opt}}=x_i(0)$；当前最优目标函数值初始化为 $f_{\text{opt}}$。

步骤 2：将 $x_i$ 映射到优化变量的取值区间为

$$z_i=a_i+(b_i-a_i)x_i\quad i=1,2 \tag{3.3.50}$$

步骤 3：对优化变量进行优化搜索，若 $f(z_i)\leqslant f_{\text{opt}}$，则 $f_{\text{opt}}=f(z_i)$，$x_{i\text{opt}}=x_i$；否则继续。

步骤 4：$n_1=n_1+1, x_i(k+1)=\mu x_i(k)[1-x_i(k)]$。

步骤 5：若经过 $M_1$ 次搜索 $f_{opt}$ 保持不变，则令 $x_i=x_{iopt}+\Delta x_i$，$\Delta x_i$ 为一个很小的数；$n_2=n_2+1$。

步骤 6：重复步骤 2～步骤 4。若 $k>M_2$，则将 $x_{iopt}$ 作为最优的混沌变量，所对应的 $z_i$ 为 SVM 优化参数。

混沌优化 SVM 参数的流程，如图 3.11 所示。

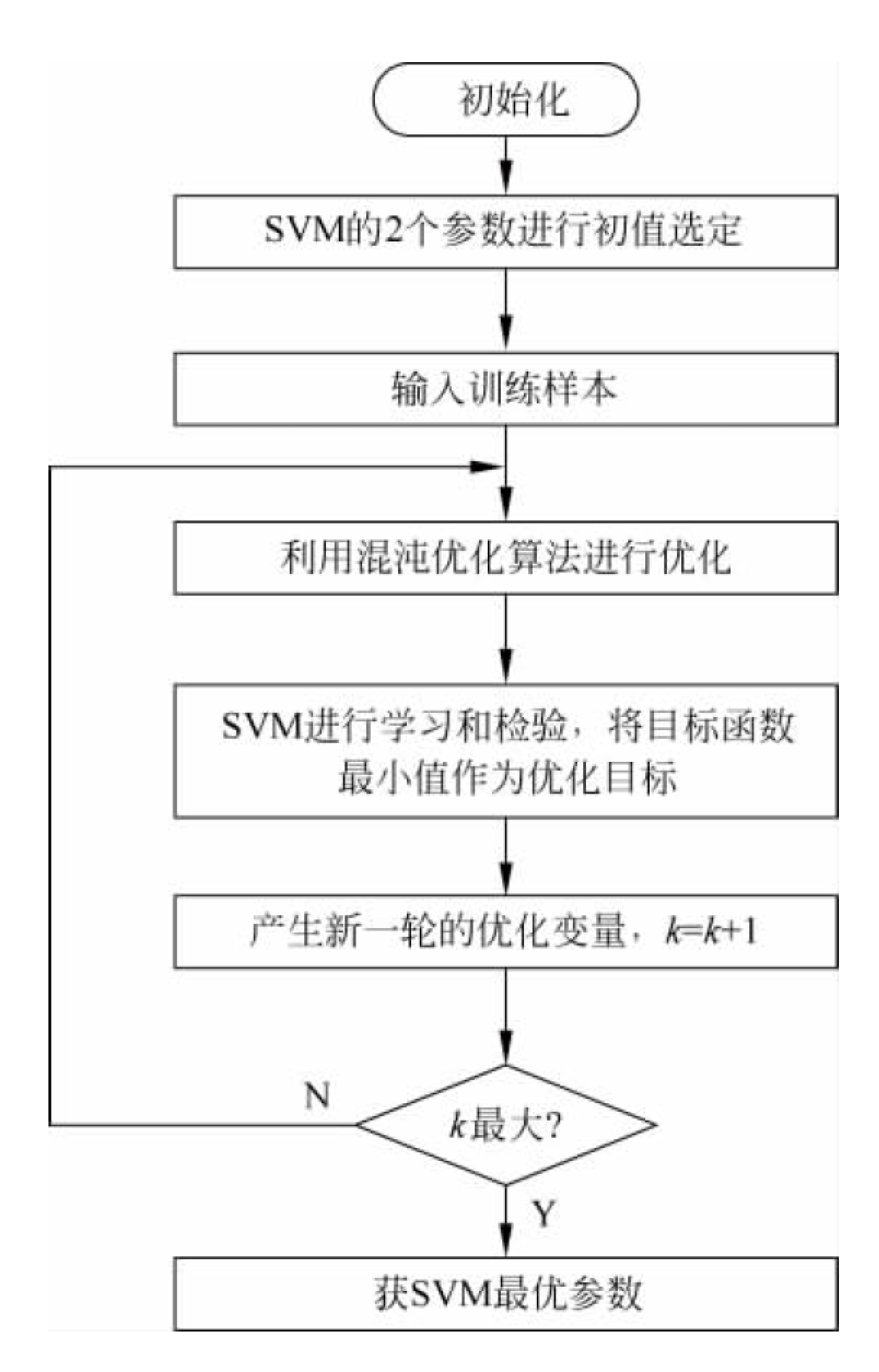



图 3.11 混沌优化 SVM 参数的流程

为提高支持向量机的拟合能力，利用混沌优化算法优化支持向量机参数；为避免正交小波多模盲均衡算法的权向量陷入局部极小值点，利用支持向量机对正交小波多模盲均衡算法的权向量进行初始化，最终得到了 CSVM-WTWMMA。

### 3.3.5　算法仿真

**【实验 3.2】**　为了验证 CSVM-WTWMMA 的有效性，用水声信道进行仿真研究，并与 WTWMMA，WMMA 进行比较。

仿真实验中，采用混合水声信道[0.3132　−0.1040　0.8908　0.3134]，信噪比为 30dB，均衡器的权长为 16。

发射信号为 128QAM，步长因子 $\mu$ 分别为 $\mu_{\mathrm{WMMA}}=3.8\times10^{-7}$，$\mu_{\mathrm{WTWMMA}}=3.5\times10^{-5}$，$\mu_{\mathrm{CSVM\text{-}WTWMMA}}=3.2\times10^{-5}$；$M_1$、$M_2$ 的值分别是 300、100，$\Delta x_i$ 的值都为$10^{-3}$；$z_1$ 的遍历区间为$(0,C]$，$z_2$ 的遍历区间为$(0,\varepsilon]$，$C$、$\varepsilon$ 的值由式(3.3.41)和式(3.3.42)确定；对信道的输入信号采用 DB2 正交小波分解，分解层次是 2 层，功率的初始值为 4，遗忘因子 $\beta=0.99$；加权因子 $\lambda=0.78$；采用均衡器输入数据的前 300 点对权向量进行初始化，初始化的切换条件 $\zeta$ 为 $10^{-5}$。在保证眼图完全睁开的前提下，800 次蒙特卡罗仿真结果如图 3.12 所示。

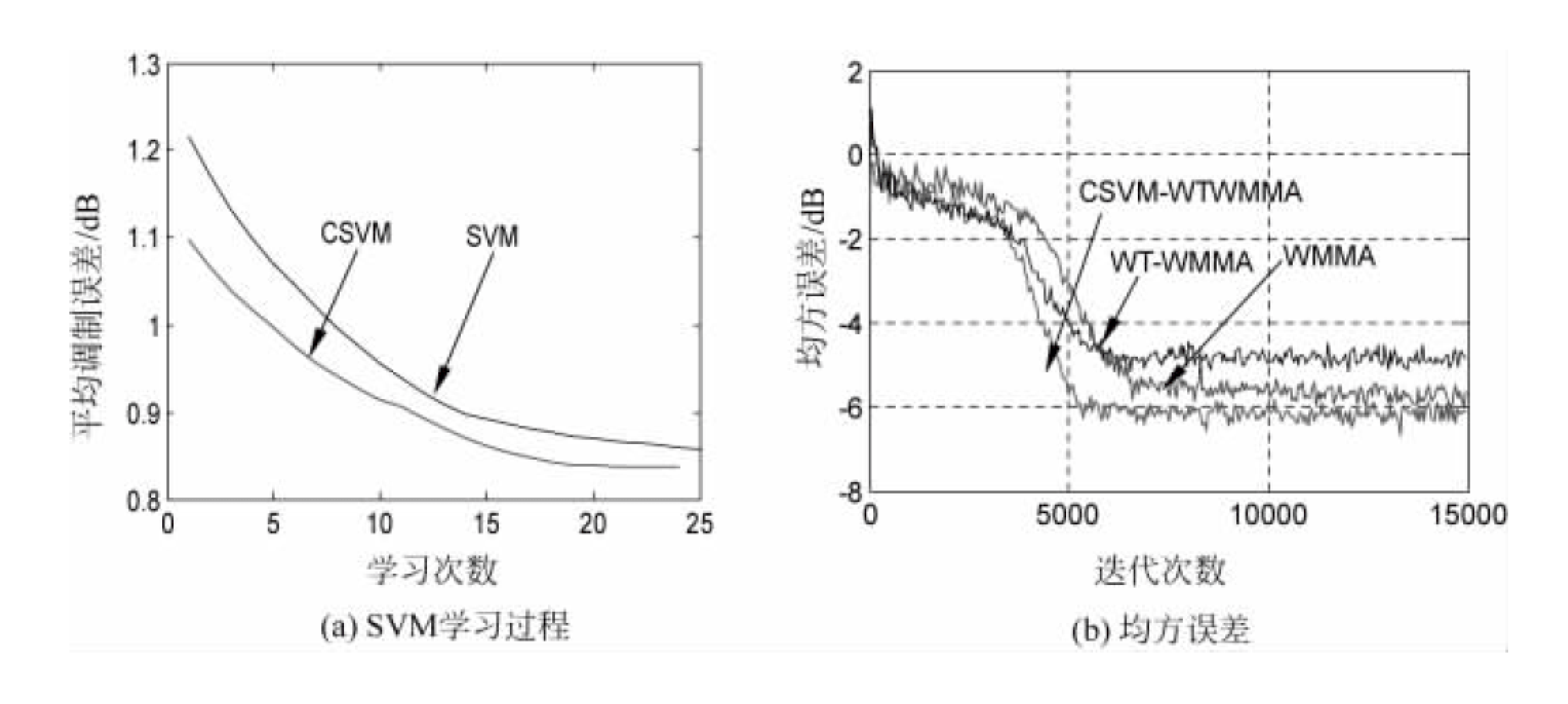



图 3.12　仿真结果

图 3.12(a)表明，CSVM 在每次学习过程中，平均调制误差比参数优化前的 SVM 小约 0.05dB；图 3.12(b)表明，CSVM-WTWMMA 收敛后，均方误差比 WTWMMA 约小 0.2dB，比 WMMA 约小 1dB；收敛速度比 WTWMMA 快约 1000 步，比 WMMA 快了约 1200 步。

# 参 考 文 献


[1] 乐波.基于混沌振子与小波的低信噪比信号检测研究与应用[D].成都：电子科技大学,2009.

[2] Erdmann D, Murphy S. Honen stream cipher[J]. Electronics Letters,1992,28(9):893-895.

[3] 李辉.混沌数字通信[M].北京：清华大学出版社,2006.

[4] Lixin Yang, Zhongrong Zhang, Jiangang Zhang. Using a neural network controller to control chaos in the rossler systems[J]. Artificial Intelligence and Computational Intelligence,2009,3:50-53.

[5] Solak E. Partial identification of Lorenz systems and its application to key space reduction of chaotic cryptosystems[J]. Circuit and Systems Ⅱ:Express Briefs,2004,51(10):557-560.

[6] Lima J B, Panario D,Qiang Wang. A Karatsuba-based algorithm for polynomial multiplication in Chebyshev Form[J]. IEEE Journals & Magazines,2010,59(6):835-841.

[7] 陈双,郭建勤.混沌优化算法在组合优化问题中的应用[J].现代电子技术,2008,31(18):68-70.

[8] 孙静.基于混沌理论的正交小波变换盲均衡算法[D].淮南：安徽理工大学,2012.

[9] 徐文才.基于混沌技术的水声通信系统[D].南京：南京信息工程大学,2012.

[10] 郭业才,孙静.基于混沌系统的正交小波变换盲均衡算法[J].控制工程,2012,19(3):443-447.

[11] Yang J, Werner J,Dumont G A. The multimodulus blind equalization and its generalized algorithm[J]. IEEE Journal on Selected Areas in Communications,2002,20(5):997-1015.

[12] Fan C P, Liang W H,Lee W. Fast blind equalization with two-stage single/multilevel modulus and DD algorithm for high order QAM cable systems[C]. Proc of ISCAS 2008,2008:3001-3004.

[13] Hel,Amin G, Reed C,et al. A hybrid adaptive blind equalization algorithm for QAM signal in wireless communications[J]. IEEE Trans on Signal Processing,2004,52(7):2058-2069.

[14] 王彬,葛临东,吴迪.适用于高阶 QAM 信号的盲均衡算法[J].通信学报,2010,31(8A):10-16.

[15] 许小东,戴旭初,徐佩霞.适合高阶 QAM 信号的加权多模盲均衡算法[J].电子与信息学报,2007,29(6):1352-1355.

[16] Santamaria I, Ibanez J,Vielva L,et al. Blind equalization of constant modulus signals via support vector regression[C]. Proceeding of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Hong Kong,2003,2:737-740.

[17] V Vapnik. The Nature of Statistical Learning Theory[M]. New York: Springer-Verlag,1995.

[18] V Cherkassky, Y Ma. Selection of meta-parameters for support vector regression[C]. Proceedings of International Conference on Artificial Neural Networks(ICANN),Berlin,Germany,2002:687-693.

[19] Chang Shuang, Guo Jian-qin. Application of chaos optimization algorithm in the solution of combination optimization problems[J]. Modern Electronic Technique,2008,31(18):68-70.

[20] Guo Li-hua, Tang Wen-cheng,Zhan Chun-hua. A new hybrid global optimization algorithm based on chaos search and complex method[C]. IEEE International Conference on Computer Modeling and Simulation,2010,3:233-237.

[21] 董春曦,饶鲜,杨绍全.支持向量机参数选择方法研究[J].系统工程和电子技术,2004,26(4):1117-1120.

[22] V ladimir Cherkassky, Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression[J]. Neural Networks,2004,17(1):113-126.

[23] 郭业才,徐文才,许芳.混沌支持向量机优化小波加权多模盲均衡算法[J].系统仿真学报,2013,25(3):451-456.

[24] S Chen, S Gunn,C J Harris. Decision feedback equalizer design using support vector machines[J]. IEEE Proc. Vis. Image Signal Process.,2000,147(3):213-219.

[25] D J Sebald, J A Bucklew. Support vector machine techniques for nonlinear equalization[J]. IEEE Trans. On Signal Processing,2000,48(11):3217-3226.

[26] F Perez-Cruz, A Navia-Vazquez,P L Alacon-Diana,et al. SVC-based equalizer for burst TDMA transmissions[J]. Signal Processing,2001,81:1681-1693.

# 第 4 章　基于免疫克隆算法优化的盲均衡算法

**【内容导语】**　本章在阐述免疫克隆算法的基本思想、基本概念及实施步骤基础上，对影响算法性能的参数进行了分析。用免疫克隆算法对常模盲均衡算法、正交小波盲均衡算法及正交小波支持向量机常模和多模盲均衡算法进行了优化。在分析自适应免疫克隆算法基础上，将自适应免疫克隆算法引入到正交小波超指数迭代盲均衡算法中，提高了算法性能。受多种遗传算法思想启发，提出了多种群免疫克隆算法，并用其对正交小波多模盲均衡算法进行了优化。

## 4.1　免疫克隆算法

### 4.1.1　免疫克隆算法的基本思想

免疫克隆选择算法（immunity clonal selection algorithm，ICSA）是模

拟生物免疫系统的克隆选择原理而设计出来的一种启发式多峰值搜索算法[1-3]。1959年，著名的澳大利亚免疫学家Burnet提出了克隆选择原理，并说明了只有识别抗原的细胞才被选择并进行克隆扩增；由Castro等人于2000年对免疫理论中的克隆选择机理进行浓缩和概括后所提出一种崭新的启发式智能优化算法，其主要具有多样性、高变异性和依据亲和度成熟(affinity maturity)的特点，能在一定程度上解决以往进化算法通常面临的早熟收敛问题，也能在全局搜索范围内以较快的收敛速度搜索。在克隆选择算法中，抗体对应于优化问题的可能解，抗原对应于优化问题的目标函数，称其为亲和度函数。以物竞天择、适者生存的遗传法则为基础，选择亲和度函数值(简称亲和度值)高的抗体进行克隆，对克隆后的抗体按一定概率执行超变异操作。将生物免疫应答中的进化链(抗体群→免疫选择→细胞克隆→高频变异→产生新抗体→新抗体群)抽象为数学上的进化寻优过程，使抗体逐步形成包含有近似最优解的状态。

### 4.1.2 免疫算法的基本概念

免疫算法是基于免疫学理论和生物免疫系统机制而提出的，是对生物免疫机理的一种模拟。为了便于理解，需了解免疫系统中几个重要概念。

1. 抗原

在免疫系统中，抗原(antigen，Ag)一般是问题目标函数的函数。在优化问题中，抗原定义为问题的优化目标(目标函数)及其约束条件。

2. 抗体

一般将待求解问题解空间中的一个解对应于免疫算法中的一个抗体(antibody，Ab)，与遗传算法中的个体相似，抗体的集合称为抗体群。抗体一般是以二进制(取海明距离)和十进制(取为欧几里得距离)两种编码形式出现的。

3. 亲和度

亲和度(affinity)(结合力的大小称为亲和度)包括抗体对抗原的亲和度(解所对应的目标函数值),以及抗体和抗体之间的亲和度(在解空间中,两个解之间的距离)两种类型。

4. 克隆

克隆(clone)实质上就是在抗体进化过程中,根据亲和度值大小进行克隆,即生物通过无性繁殖连续传代的增生过程。

抗体群通过克隆选择操作不断进化,会得到一个新的抗体群,其抗体群的演化过程为

$$A(K)\xrightarrow{\text{克隆操作 } T_c^C} Y(K)\xrightarrow{\text{免疫基因操作 } T_g^C} A''(K)\cup A'(K)\xrightarrow{\text{克隆选择操作 } T_s^C} A(K+1)$$

5. 单克隆抗体

机体淋巴组织内可存在多种抗体形成细胞(B细胞),当受刺激后,对应一个抗原决定簇,每种B细胞可增殖分化为一种细胞群,并且细胞群能分泌合成在理化性质、分子结构、遗传标记、生物学特性等方面均有相同特性的抗体,则称这种抗体为单克隆抗体(monoclonal antibody)。

单克隆选择算法(single clonal selection algorithm,SCSA)[3]的缺陷:单克隆选择算法虽然克服了免疫算法局部搜索能力差且进化缓慢的问题,但用较小的变异概率来提高局部搜索能力,降低了全局搜索能力;而较大的变异概率虽然全局寻优能力得到提高,但收敛的精度下降,因此,其全局搜索能力和局部搜索能力是矛盾的;同时,该算法没有体现抗体与抗体之间的相互作用关系,这导致算法搜索过程中出现群体多样性的缺乏及抗体的选择不具有随机性,即仅有适应度改善的抗体或克隆被保存,这隐含着算法可能陷入局部收敛而不能收敛到全局最优,而且也没有对算法进行理论和定量分析;在SCSA中,所有个体都是二进制编码,计算时需要将十进制

数转化为二进制数，最后又必须将二进制数再转化为十进制数，这对多维函数优化问题，二进制编码面临“维数灾”问题。

6. 多克隆抗体

天然抗原物质往往具有多种不同的抗原决定簇，而每一决定簇都可以刺激机体，一种抗体形成细胞产生一种特异性抗体(单克隆抗体)。这种多种抗原决定簇可刺激多种细胞克隆合成分泌各种不同的抗体，即多克隆抗体(polyclonal antibody)。

多克隆：抗体克隆选择学认为抗体是天然产物，以受体的形式存在于细胞表面，抗原可与之选择性地反应。当抗原侵入生物体时，其免疫系统在机体内选择能识别和消灭相应抗原的抗体，主要借助克隆使之激活、分化和增殖，以增加该抗体的数量，进行免疫应答(immune response)以最终清除抗原。在这一过程中，多克隆是免疫应答特异性的基础，与单克隆只持续表达抗体一种或少数几种抗原决定簇和表位不同，多克隆性在细胞水平上的表现是TCR和BCR结构(B细胞受体即BCR，T细胞受体即TCR)的极端多样性，因此直接导致了抗体网络的多样性、记忆性和特异性。多克隆算子正是模拟生物免疫系统的多克隆机理，采用变异来实现抗体间的信息交换，而且还要充分利用抗体间的协作，采用交叉来促进不同抗体间信息的交流，有利于增加种群多样性，提高种群的收敛速度。单克隆算子包括三个步骤，即克隆、克隆变异和克隆选择；而多克隆算子包括克隆、克隆变异、克隆交叉(重组)和克隆选择四个步骤。多克隆算子不仅有利于增加种群多样性，还能提高种群的收敛速度。因此，用多克隆算子作为新型克隆选择算法的主要操作算子，为利用各种有效的交叉、变异方法提供了广阔的空间[3]。

### 4.1.3 生物免疫系统的特性

1. 自适应性

免疫系统能学习识别和应答新的抗原，同时，通过免疫记忆保持识别以

前见过的抗原的能力，因此具有自适应性。自然界中存在的抗原种类远远多于生物体内的抗体种类，因此侵入生物体内的抗原具有不可预知性。但免疫系统会通过免疫细胞的增殖和分化过程，不断地产生新的抗体，最终生成适合的抗体来消灭抗原，从而动态地适应外界环境的变化，两者相互竞争共同进化。同时，在对系统其他部分的刺激和与环境交互做出应答时，免疫系统能够忽略那些无用的或者高危的元素，并能提高现存元素的性能。我们借鉴免疫应答的动态适应性，采用尺度收缩因子，以达到抗体搜索范围随进化代数能够自适应的变化；同一代群体中的抗体依据亲和度大小自适应调整每个抗体交叉概率、变异概率和克隆数。

2. 多层性

生物体的免疫分为非特异性免疫和特异性免疫两个层次。非特异性免疫作用具有广泛性，即这种抗击病原体的能力不是专门针对某一种病原体，而是对许多种病原体都能起到同样的抗击作用，因此称为非特异性免疫。特异性免疫是生物体免疫系统产生一些淋巴细胞（包括 T 细胞和 B 细胞）去识别和对抗侵入机体的病原体，识别成功后还可以将这种病原体的特征长期记忆下来，同样的病原体再次侵入时将会被很快识别和消灭；这是专门针对一种病原体的识别和杀灭作用。这两种免疫系统各司其职、相互影响，使人体的整个免疫系统成为一个多层次的结构。根据生物免疫系统多层次结构的机理，新型克隆选择算法将抗体群分为四类，即死亡抗体群、普通抗体群、记忆抗体群和混沌抗体群，对不同抗体群采用不同的操作算子。

3. 分布性

免疫系统能够检测分布在机体内部各个部位的抗原。同时，免疫系统是由局部相互作用的基本单元来提供全局的保护，没有中央控制器，它由广泛分布于全身的免疫细胞组成，这些免疫细胞通过在时间和空间上的分布式网络结构来实现各种免疫功能。这意味着免疫系统具有高度分布性。免疫系统分布式特性可以把工作载荷分布到不同的多个工作单元上，能够有

效地提高系统的工作效率；同时，还可以减少由局部工作单元失效引起的对系统整体的不利影响。

4. 多样性

在胚胎期，由于遗传和免疫细胞在增殖中发生基因突变，形成了免疫细胞的多样性。首先，机体内免疫细胞的多样性保证了当任意一种抗原侵入机体时，免疫系统都能找到可识别和消灭此种抗原的免疫细胞，并使之激活、分化和增殖，进行免疫应答，最终清除抗原。其次，每个个体都有一个独一无二的免疫系统，从而使整个群体的免疫系统多样性。这种多样性保证了当有个体对某抗原呈现脆弱性时，不会出现所有个体都对同一抗原呈现脆弱性的情况。因此，免疫系统的多样性可大大增强个体和群体健壮性。

### 4.1.4 免疫克隆算法的实现步骤

1. 抗体的编码

编码就是将一个实际问题的所有解，从其解空间转换到免疫克隆算法所能处理的搜索空间。对抗体种群的编码方式目前常用的有二进制编码、自适应编码、实数编码或者乱序编码。本章采用二进制编码方式对抗体进行编码，问题空间的参数由二值符号集{0,1}组成。

2. 初始种群的产生

在已编码的解空间中，随机产生 $N$ 个初始解，每个初始解为一个抗体，$N$ 个抗体组成一个种群称为抗体种群。免疫克隆选择算法以这个初始的抗体种群作为起点开始进行迭代。初始种群数量 $N$ 又称为种群规模，它的取值大小直接影响免疫克隆选择算法的性能；当初始种群规模过大时，虽然降低了陷入局部收敛的可能性，但算法每一次迭代的计算复杂度也随之增加，从而影响算法性能；当初始种群规模过小时，增加了迭代次数，使免疫克隆算法的搜索空间受限，使算法陷入局部最优。因此，在选取种群规模

大小时，既要保持种群的多样性，又要兼顾算法的性能。所以，在具体应用时，应该根据不同问题要求，调节好计算速度和计算精度之间矛盾，进行多次试验来确定一个合适的种群规模。

3. 亲和度函数的确定

在一般优化问题中，通常将衡量优化的指标称为目标函数，该步骤主要是对亲和度函数进行计算。亲和度函数一般是抗原和抗体之间结合强度的函数，是反映抗体优劣程度的一种评价值，是抗体种群进化发展的重要指标，所以要对亲和度函数进行合适的定义。在克隆选择算法中，抗体对应于优化问题的可能解，抗原对应于优化问题的目标函数，即亲和度函数。亲和度值越大，则抗体识别抗原的能力越大。

为了使免疫克隆选择算法中亲和度函数与抗体群中的个体优劣度量相互联系，将其亲和度函数对应于问题的目标函数。一般最优化问题可分为两类：一类是要求为非负，而且越大越好，用于求问题的极大值；另一类是对于所给定的优化问题的目标函数 $J(x)$ 可能为正也可能为负，并且可能是求问题的极小值，因此对目标函数 $J(x)$ 进行函数变换，保证目标函数的优化方向对应于亲和度值增大的方向且其值为非负。

对于最小化问题，则亲和度函数 $\mathrm{Fit}(x)$ 和目标函数 $J(x)$ 间的映射关系为

$$\mathrm{Fit}(x)=\begin{cases}J_{\max}-J(x), & J(x)<J_{\max}\\ 0, & \text{其他}\end{cases} \tag{4.1.1}$$

式中，$J_{\max}$ 为 $J(x)$ 的最大值估计。

对于最大化问题，则亲和度函数 $\mathrm{Fit}(x)$ 和目标函数 $J(x)$ 间的映射关系为

$$\mathrm{Fit}(x)=\begin{cases}J_{\max}+J(x), & J_{\max}+J(x)>0\\ 0, & \text{其他}\end{cases} \tag{4.1.2}$$

4. 克隆操作 $T_c^C$

令 $\boldsymbol{A}(k)=[A_1(k),A_2(k),\cdots,A_n(k)]$ 表示第 $k$ 代的抗体群。克隆操作

的实质就是将抗体 $\boldsymbol{A}(k)$复制,使一个抗体克隆出多个抗体。通常克隆的个数定义与抗体亲和度(或抗体亲和度在抗体种群中的排名)成正比,与抗体-抗体亲和度成反比,以提高收敛速度,维护群体多样性,避免早熟收敛。

设克隆操作前的抗体种群为 $\boldsymbol{A}(k)$,克隆操作后的抗体种群为 $\boldsymbol{Y}(k)$,克隆操作 $T_c^C$ 定义为

$$\boldsymbol{Y}(k) = T_c^C(A(k)) = [T_c^C(A_1(k)) \quad T_c^C(A_2(k)) \quad \cdots \quad T_c^C(A_n(k))] \tag{4.1.3}$$

式中,$Y_i(k)=T_c^C(A_i(k))=I_iA_i(k)$,$i=1,2,\cdots,n$,$I_i$ 为元素为 1 的 $q_i$ 维行向量,$q_i\in[1,B_i]$称抗体 $A_i$ 的 $q_i$ 克隆。

$$q_i(k) = \operatorname{Int}\left(N * \frac{\operatorname{Fit}(A_i(k))}{\sum_{j=1}^{n}\operatorname{Fit}(A_j(k))}\right) \tag{4.1.4}$$

$$B_i = \operatorname{round}\left(\alpha \frac{\operatorname{Fit}(A_i)}{\sum_{i=1}^{N}\operatorname{Fit}(A_i)} + b\right) \tag{4.1.5}$$

式中,$\alpha$ 为$\geqslant 1$ 的比例系数;$i=1,2,\cdots,n$;round(·)取整函数;$b$ 为$>1$ 的常量。为了保证每个抗体都有一定克隆数量,故加上了常量 $b$;$N$ 是与克隆总规模有关所设定的参数值,且满足 $N>n$;Int(·)为取正函数,Int($x$)表示取大于 $x$ 的最小整数。由式(4.1.4)知,亲和度 Fit($A_i(k)$)越大,规模 $q_i$ 就越大,反之就小。

克隆扩增后的抗体种群变为

$$\boldsymbol{Y}(k) = [Y_1(k) \quad Y_2(k) \quad \cdots \quad Y_n(k)] \tag{4.1.6}$$

式中,$Y_i(k)=\{Y_{ij}(k)\}=[Y_{i1}(k)Y_{i2}(k)\cdots Y_{iq_i}(k)]$,$Y_{ij}(k)=A_{ij}(k)=A_i(k)$,$j=1,2,\cdots,q_i$。

### 5. 免疫基因操作 $T_g^C$

免疫基因主要包括变异和交叉。根据生物学中单、多克隆抗体对信息交换多样性特点,将仅采用变异的克隆选择算法定义为单克隆选择算法(monoclonal selection algorithm,MCSA),而采用交叉和变异的定义为多

克隆选择算法(polyclonal selection algorithm,PSA)。

1) 变异

免疫基因变异定义为

$$Z(k) = T_g^C(Y(k)) \tag{4.1.7}$$

(1) 几种常见变异算子

① 抗体交换算子

抗体两点变换算子是指抗体按照一定的交换概率 $p_c$,随机选取抗体中的两个或多个点,并交换这些点上的基因形成新的抗体,如图 4.1 所示(两点变换)。

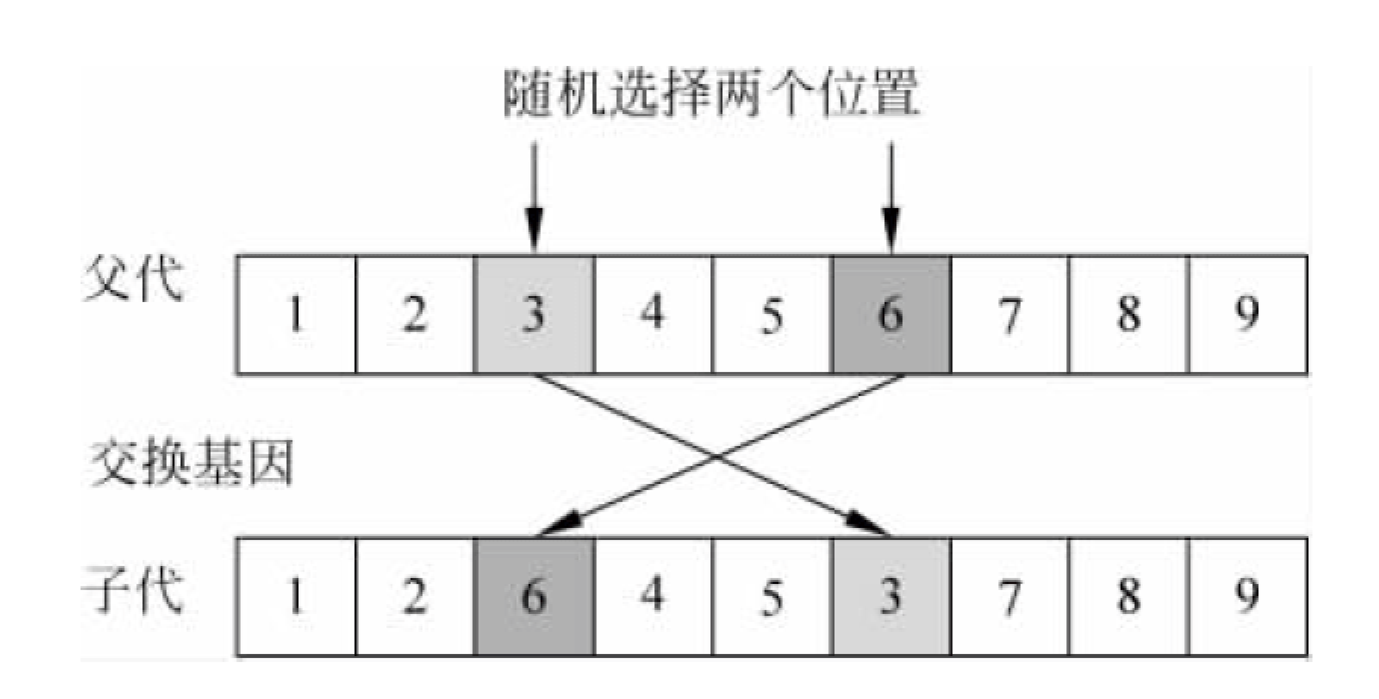



图 4.1　抗体交换算子

② 抗体倒位算子

抗体倒位算子是指抗体按照一定的逆转概率,随机选取抗体中的两个点,将这两个点之间的基因段首位依次倒转过来而构造新的抗体,如图 4.2 所示。

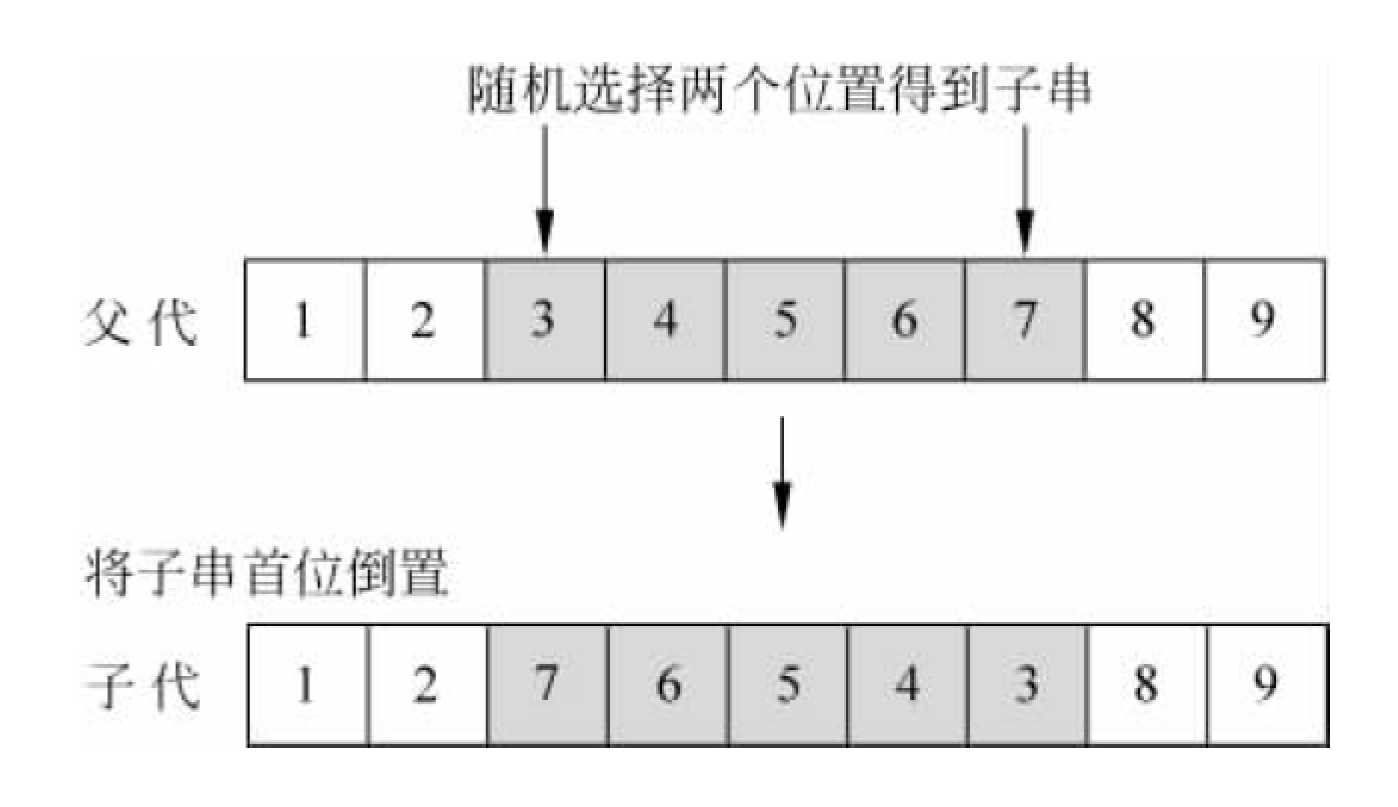



图 4.2　抗体倒位算子

③ 抗体移位算子

抗体移位算子是指抗体按照一定的移位概率 $p_s$，随机选取抗体中的两个点，将两点之间基因段中的基因循环依次向左或向右移动，使该基因段中的末位基因移到段的首位而形成新的抗体，如图 4.3 所示。

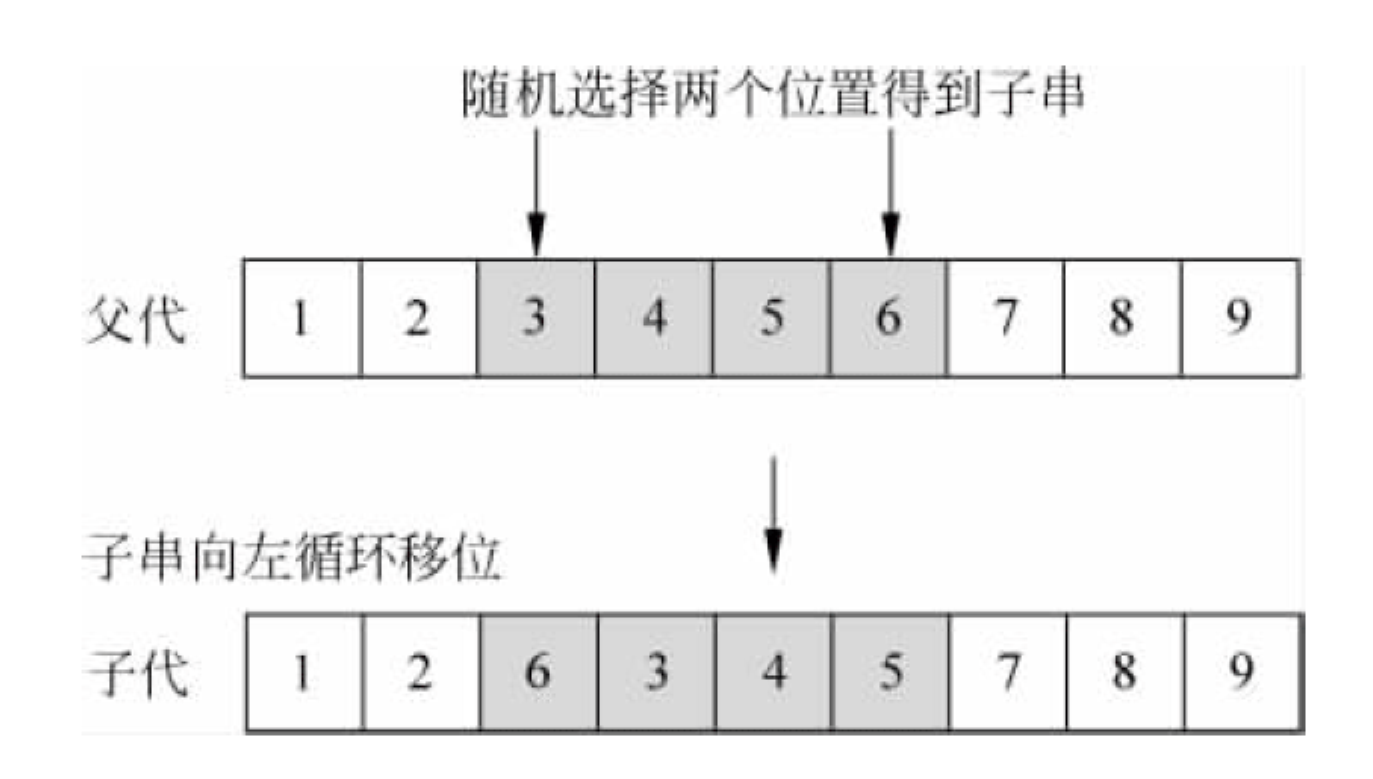



图 4.3　抗体移位算子

④ 抗体变异算子

抗体变异算子是指抗体按照一定的变异概率 $p_m$，随机选取抗体中的一个或多个点，并由随机生成的一个或多个基因位来取代，以形成新的抗体，如图 4.4 所示(单点变异基因)。

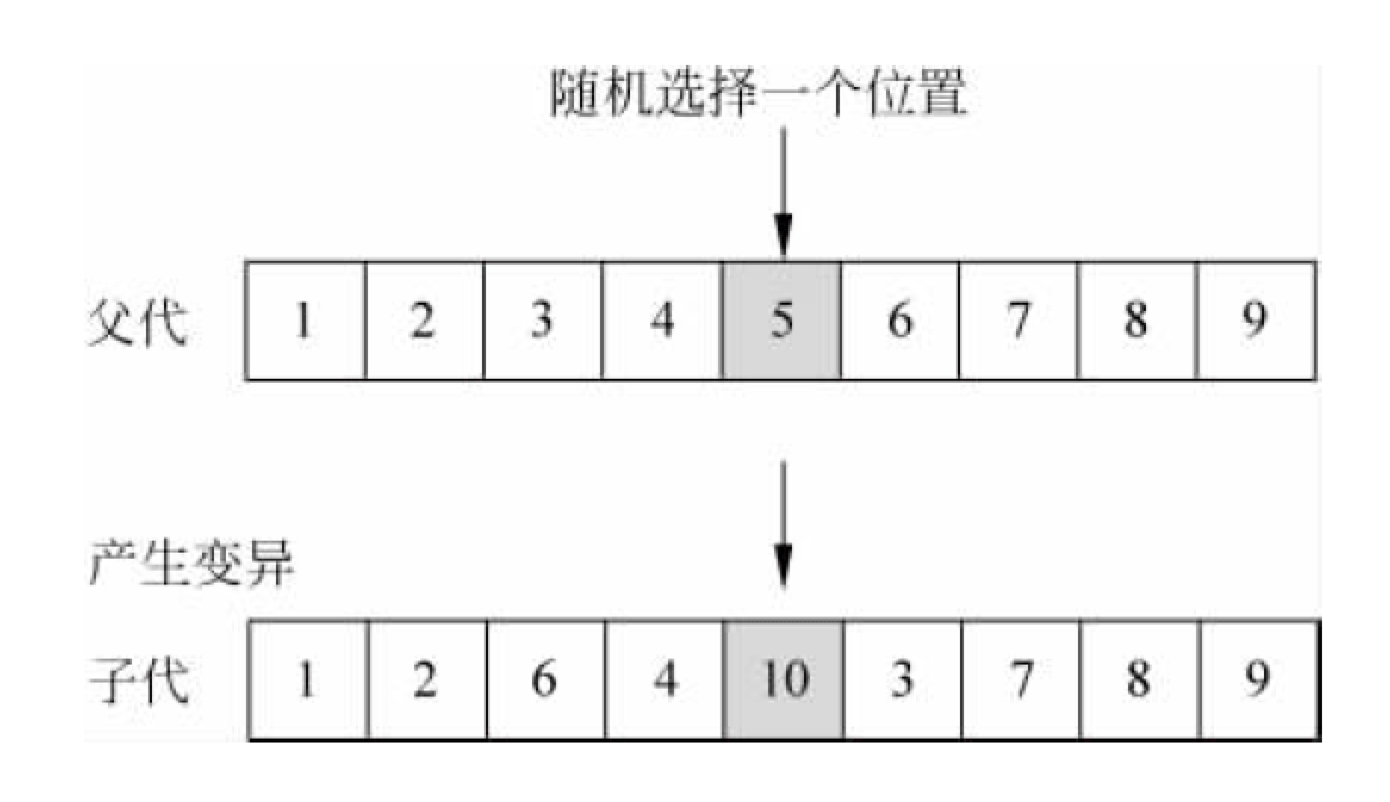



图 4.4　抗体变异算子

⑤ 均匀与非均匀变异算子

均匀变异算子是指分别用符合给定区间内均匀随机分布数，以某一较小的概率来改变个体编码串中各个基因座上的原有基因座。

假设有一个抗体 $x=(x_1,x_2,\cdots,x_k,\cdots,x_n)$。若 $x_k$ 为变异点，其取值

范围为$[V_{\min}^{k},V_{\max}^{k}]$，在该点对个体 $x$ 进行均匀变异后，得到一个新的抗体 $x=(x_1,x_2,\cdots,x_k',\cdots,x_n)$，其中变异点的新基因值为

$$x_k' = V_{\min}^{k} + \text{rand}\cdot(V_{\max}^{k} - V_{\min}^{k}) \tag{4.1.8}$$

式中，rand 为区间[0,1]内符合均匀分布的一个随机数。

非均匀变异算子操作过程跟均匀变异相似，但它重点搜索空间在原个体邻域内的微小范围内进行微调。当由个体 $x=(x_1,x_2,\cdots,x_k,\cdots,x_n)$ 向 $x=(x_1,x_2,\cdots,x_k',\cdots,x_n)$ 的非均匀变异操作时，若变异点 $x_k$ 处的基因值取值范围为$[V_{\min}^{k},V_{\max}^{k}]$，则新的基因 $x_k'$ 产生公式为

$$x_k' = \begin{cases} x_k + \Delta(t, V_{\max}^{k} - x_k), & \text{rand} = 0 \\ x_k + \Delta(t, x_k - V_{\min}^{k}), & \text{rand} = 1 \end{cases} \tag{4.1.9}$$

式中，$\Delta(t,y)$（$y$ 表示 $V_{\max}^{k}-x_k$ 或 $x_k-V_{\min}^{k}$）表示$[0,y]$范围内符合非均匀分布的一个随机数，要求随着进化代数 $t$ 的增加，$\Delta(t,y)$接近于 0 的概率也逐渐增加。例如，$\Delta(t,y)$可按 $\Delta(t,y)=y(1-(\text{rand})^{(1-\text{rand}/T)b})$ 决定，其中，$T$ 是最大进化代数，$b$ 是一个系统参数，它决定了随机数扰动对进化代数 $t$ 的依赖程度。

（2）高频变异

本章根据文献[4]提出方法，采用高频变异。

高频变异算子作为克隆选择的主要操作算子，能够维持亲和度成熟和抗体多样性的产生，扩大搜索范围，防止出现早熟现象，对算法局部搜索能力和收敛性有着重要的影响。根据抗体亲和度值大小而实行不同的变异，具体操作为对高频变异的基因位按式(4.1.10)及式(4.1.11)进行操作得到变异集。

$$X = x + \alpha \cdot N(0,1) \tag{4.1.10}$$

$$\alpha = (1/\gamma)\cdot \exp(-\text{Fit}) \tag{4.1.11}$$

式中，$X$ 表示变异体；$N(0,1)$表示均值为 0、标准方差为 1 正态随机变量；$\alpha$ 表示变异概率系数；$\gamma$ 表示变异的控制系数；Fit 表示抗体与抗原的亲和度值。

依据变异概率 $p_m$ 对克隆后的群体中第 $i$ 个抗体进行变异操作，即

$$Z_i(k) = T_g^c(Y_i(k)), \quad 1 \leqslant i \leqslant n \tag{4.1.12}$$

一般情况下，采用较大的变异概率使算法在问题求解中随机搜索，而较小的变异概率可以有效防止丢失种群中重要的、单一的基因。通过利用免疫基因操作中高频变异操作，在单一抗体周围产生一个变异解的群体，利用局部搜索提高抗体与抗原之间的亲和度，更好地增加了解的多样性。

(3) 动态变异概率

高频变异概率只在算法执行的初期有效，当算法运行到后期尤其将收敛到近似解时，仍通过高频变异概率变异得到更优抗体的概率显得极低，这就需要动态地降低变异算子的变异概率以尽可能提高收敛速度。动态变异概率定义为

$$p = p(1 - t \cdot \alpha / T) \tag{4.1.13}$$

式中，$p$ 为变异概率；$t$ 为当前进化代数；$T$ 为总进化代数；$\alpha$ 为常数。

2) 交叉

本章根据文献[4]提出的方法，采用精英交叉进行免疫基因操作。

精英交叉原理如下：在免疫算法中，首先给定一个精英交叉概率 $p_{kc}$（kc 表示 king-crossover，即精英交叉），对于克隆抗体群中第 $t$ 代每个个体 $a(t)$ 产生一个[0,1]之间的随机数 rand，如果 rand 小于精英交叉概率 $p_{kc}$，则 $a(t)$ 被选中与保存的当前代精英个体 $b(t)$ 进行交叉，其方法是：将 $a(t)$ 和 $b(t)$ 放入一个小的交配池中，根据选定的交叉策略（单点、两点、多点和一致交叉等），对 $a(t)$ 和 $b(t)$ 进行交叉操作，得到一对子代个体 $a'(t)$ 和 $b'(t)$。然后，用 $a'(t)$ 替代种群中的 $a(t)$，$b'(t)$ 则丢弃不用。

通过精英交叉，抗体群继承了精英个体的优良模式，并且不会破坏其优良模式，与传统的交叉操作相比具有更大的优越性，因为精英交叉总是能增加群体中的优良模式，而且精英交叉策略对免疫算法的运行时间、平均收敛代数、标准方差等性能有着极大的改善。

6. 克隆选择操作 $T_s^C$

克隆选择操作 $T_s^c$ 与遗传算法中的选择操作的区别在于：克隆选择操作是从各自克隆增殖后的抗体子代中选择最优的个体（无性选择过程），重新组合而形成新的种群，即

$$A_i(k+1) = T_s^c(Z_i(k) \tilde{\cup} A_i(k)) = \begin{cases} B_i(k), & \text{Fit}(A_i(k)) < \text{Fit}(B_i(k)) \\ A_i(k), & \text{其他} \end{cases} \tag{4.1.14}$$

式中，$\tilde{\cup}$ 操作就是将 $A_i(k)$ 与 $Z_i(k)$ 合并产生新的抗体种群。

新一代种群为

$$\begin{aligned} A(k+1) &= T_s^c(Z_i(k) \cup A_i(k)) \\ &= [T_s^c(Z_1(k) \cup A_1(k)) T_s^c(Z_2(k) \cup A_2(k)) \cdots T_s^c(Z_n(k) \cup A_n(k))] \\ &= [[A_1(k+1) A_2(k+1) \cdots A_n(k+1)] \end{aligned}$$

为了增加种群的多样性，克隆选择操作通常采用概率选择的方法，该方法可以使某些非优势抗体得以保留和继续进化的机会，从而进一步增加抗体群的多样性[4]。

具体地，$\forall i=1,2,\cdots,n$，有

$$B_i(k) = \max\{Z_i(k)\} = \{z_{ij}(k) \mid \max(\text{Fit}(Z_{ij})) \quad j = 1,2,\cdots,q_i\} \tag{4.1.15}$$

式中，Fit(·)为优化问题的目标函数，在克隆选择算法中成为亲和度函数。$B_i(k)$ 取代 $A_i(k)$ 成为 $A_i(k+1)$ 的概率 $p_s$ 为

$$p_s = \begin{cases} 1, & \text{Fit}(A_i(k)) < \text{Fit}(B_i(k)) \\ \exp\left(-\dfrac{\text{Fit}(A_i(k) - \text{Fit}(B_i(k)))}{\alpha}\right), & \begin{array}{c}\text{Fit}(A_i(k)) \geqslant \text{Fit}(B_i(k))\text{，且 } A_i(k) \\ \text{不是目前种群的最优抗体}\end{array} \\ 0, & \begin{array}{c}\text{Fit}(A_i(k)) \geqslant \text{Fit}(B_i(k))\text{，且 } A_i(k) \\ \text{是目前种群的最优抗体}\end{array} \end{cases} \tag{4.1.16}$$

式中,目前种群中的最优抗体是通过计算当前种群中的各个抗体的亲和度值,找出亲和度最大的抗体并作为当前种群的最优抗体。$\alpha(\alpha>0)$的取值在抗体种群多样性起决定性作用,$\alpha$ 取值越大,抗体的多样性就越好。通过局部择优,从而实现种群的压缩。按上述过程迭代运算,当到达终止条件时,就得到优化问题的最优解[5]。

Jiao[6-8]等提出的基础克隆选择算法的操作过程如图 4.5 所示。

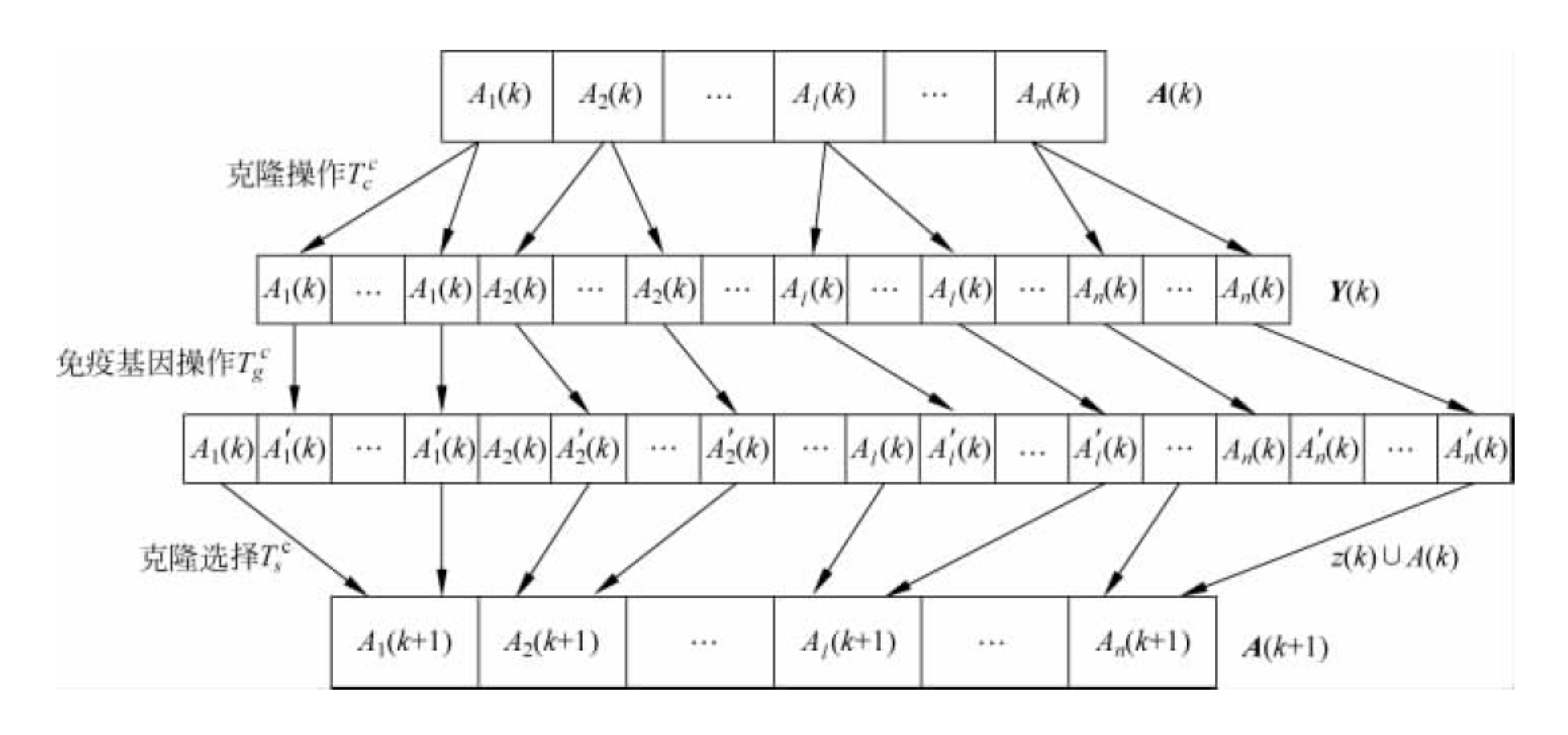



图 4.5 CSA 主要操作过程

免疫克隆选择算法流程如图 4.6 所示。

图 4.6 表明,CSA 是通过抗体空间的扩张与压缩,将局部搜索和全局搜索集合起来获得问题的最优解;换句说法,免疫克隆算法是把低维空间的问题转换到高维空间中解决,然后将结果重新投影到低维空间中而获得所求问题的满意解。

7. 影响算法性能的几个参数[9-10]

免疫算法是一种具有全局收敛能力的随机全局搜索算法,对于不同的问题或不同的算法实现形式,不同的控制参数取值会对算法性能和效率产生较大的影响。如何准确设定控制参数使算法性能得到有效提高,还依赖于免疫算法理论研究进展。

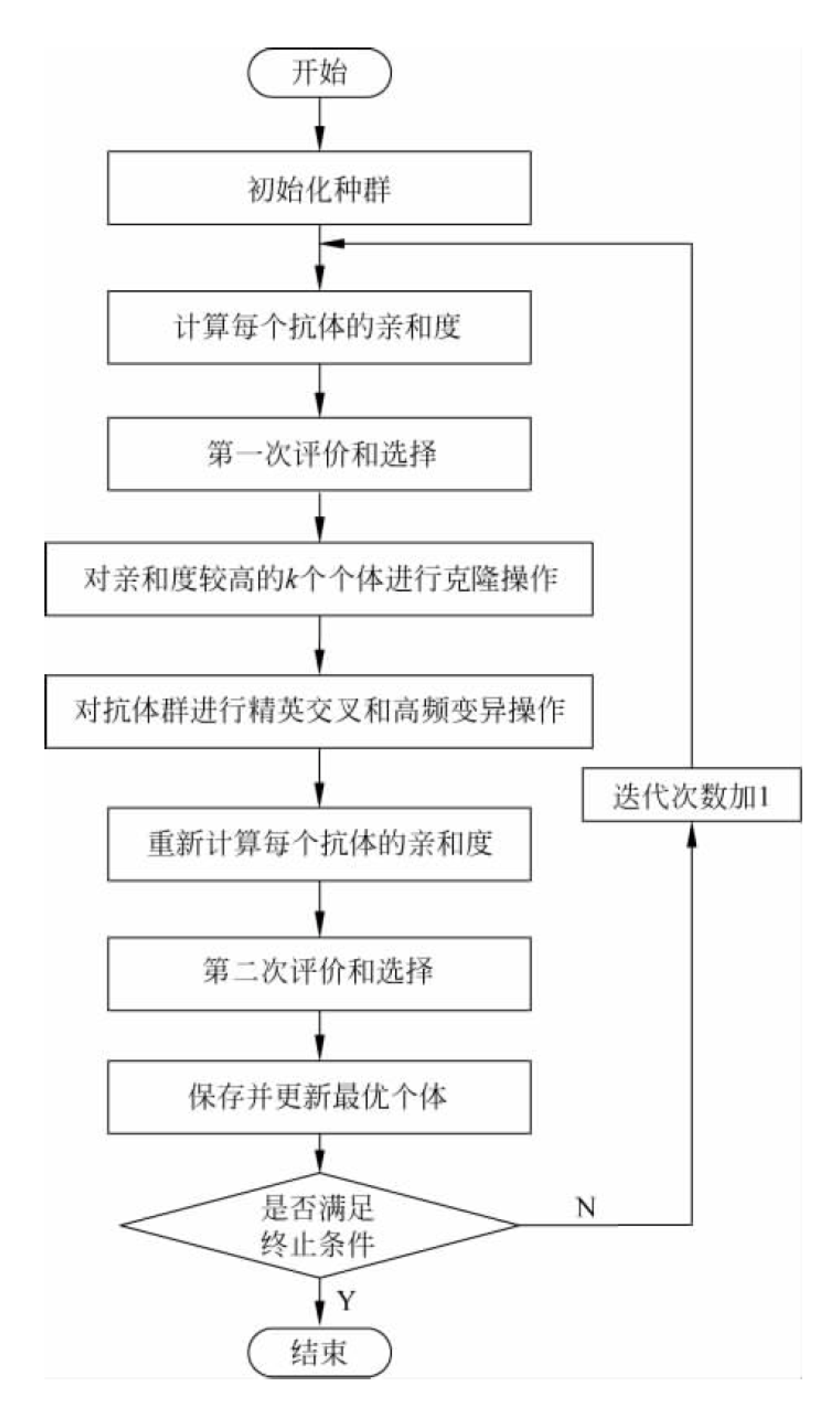



图 4.6　免疫克隆选择算法操作流程

(1) 群体规模 $N$

种群规模是随机搜索算法的一个重要参数,它是影响算法并行性的决定性因素之一。种群规模 $N$ 既不能太大也不能太小。种群规模越大,群体中个体的多样性越高,算法陷入局部解的危险就越小,但算法的计算量和搜索时间也随之增加。当群体规模太小,会使算法的搜索空间中分布范围有限,因而搜索有可能停止在早熟阶段,引起早熟收敛。显然,要避免早熟收

敛，就必须保持群体多样性，即群体规模不能太小。因此，针对不同问题的特点，种群规模的适当选择会有效提高算法的效率和性能。一般取 $N$ 为50～500。

（2）克隆系数

算法搜索能力取决于克隆扩增概率，主要是局部搜索能力，其取值越大，局部搜索能力就越好，全局搜索能力也有一定提高，但计算量也随之增大。一般取 0.5～2。

（3）交叉概率

生物基因的重组（加上变异）在自然界生物进化过程中起核心作用。同样，免疫算法中起核心作用的是抗体交叉操作。所谓交叉是指将两个父代个体中的部分结构加以替换重组而生成新个体的操作。通过交叉免疫算法的搜索能力得到明显提高，因此交叉操作是免疫算法的一个主要操作。交叉概率 $p_c$ 控制着交叉操作被使用的频度。当交叉概率较大时，虽可增强免疫算法开辟新搜索区域的能力，但可能会破坏具有高亲和力的个体结构；当交叉概率过小时，搜索过程缓慢，可能使免疫算法搜索陷入迟钝状态。一般取 $p_c$ 为 0.2～0.9。

（4）变异概率

变异操作是影响算法收敛性能的重要参数，主要是对群体中的个体串的某些基因座上的基因值作变动，使算法增加新的搜索空间、扩大搜索范围、维持群体多样性、改善局部搜索能力。免疫算法中导入变异操作主要有两个目的：①免疫算法具有局部搜索能力。当免疫算法寻优接近最优解区域时，变异操作的导入能使局部搜索能力加速向最优解收敛。这种情况下，变异概率应取较小值。②免疫算法可维持群体多样性，以防止出现早成熟收敛现象，此时变异概率应取较大值。

变异概率取值和群体规模一样，不能太小也不能太大，太小会使变异操作不明显，无法保证群体的多样性，因而不能搜索到最优值。但变异概率也不能太大，太大则使群体的稳定性变差，导致搜索过程中难以长时间稳定收敛。一般取变异率 $p_m$ 为 0.001～0.1。

（5）终止条件

由于算法无法确定当前找到的解是否是全局最优的，一般来说，将通过预先规定算法的最大进化次数来确定终止条件，也可以探测新种群中最后几代是否有改进。若没有改进，则说明寻优已趋于稳定，进化结束。一般取进化代数为100～1000。

### 4.1.5 免疫克隆算法的特点分析

免疫克隆选择算法和遗传算法一样，都属于启发式迭代优化算法，通过保留模拟生物免疫系统的若干特点来解决复杂问题，具体表现在以下几个方面。

（1）全局搜索能力

免疫克隆算法通过克隆算子将搜索空间扩展到抗体的邻域中，为了保证算法能在完整的可行解区间进行搜索，同时采用高频变异算子和消亡算子不断产生新个体，探索可行解区间的新区域，使算法具有更好的全局收敛性能。

（2）并行分布式搜索

由于免疫克隆算法中不存在对种群抗体的集中控制，种群中的每个抗体都是分布式并行进化的，因此在求问题最优解的同时还可以得到问题的多个次优解，即除了找到所求问题的最佳解决方案外，还可以得到若干个较好的备选解决方案，该特点尤其适合于解决多模态的优化问题。

（3）多样性保持机制

免疫克隆算法借鉴了生物免疫系统的多样性保持机制，计算抗体之间的亲和度，将亲和度高的抗体保留在记忆集中，同时还保留了一部分亲和度低的抗体。为了保证算法全局收敛性能，通过消亡算子产生新的随机抗体群，更好地体现抗体种群所具有多样性。

（4）鲁棒性强

免疫克隆算法的随机搜索并不针对特定的问题，对算法参数设置和初

始解的质量也没有要求，利用其启发式的智能搜索机制，即便以劣质解种群起步，最终也可以搜索到问题的全局最优解，对问题和初始解都没有依赖性，具有很强的适应性和鲁棒性。

通过分析知，免疫克隆选择算法主要通过变异和选择算子来完成进化机制的，且这两个算子都是针对局部的，即它们都是在一个抗体的克隆集中进行。当一个抗体克隆后，产生该抗体的克隆集，然后对这个克隆集进行变异和选择操作。这种操作方法虽然在每一代种群中都保留了每个抗体的历史最好情况，却导致抗体之间的信息完全没有交流，使种群失去了社会性，并且变异完全是随机和无方向的，这使算法收敛速度也相对较慢。

## 4.2 基于免疫克隆算法优化的常模盲均衡算法

免疫克隆选择算法采用群体搜索技术，不依赖于梯度信息，也不需要求解函数可微，通过用抗体群代表问题解，对抗体种群进行克隆、高频变异和选择等操作，使抗体群逐步进化到包含有近似最优解的状态。于是，本节将免疫克隆选择算法引入到常模盲均衡算法中，提出基于免疫克隆算法优化的常模盲均衡算法，水声信道仿真结果，验证了该算法的有效性。

### 4.2.1 算法原理

将免疫克隆选择算法引入盲均衡算法中，得到基于免疫克隆算法优化的常模盲均衡算法(clonal selection algorithm based CMA，CAS-CMA)[11]，如图 4.7 所示。该算法利用免疫克隆选择算法全局搜索性能来弥补 CMA 的缺陷，更好地解决搜索过程中陷于局部收敛的问题。

图中，$\boldsymbol{a}(k)$是信源信号向量或信道输入信号向量；$\boldsymbol{h}(k)$表示信道冲激响应向量；$\boldsymbol{w}(k)$表示加性高斯白噪声；$\boldsymbol{y}(k)$是均衡器输入信号向量或信道输出含噪向量；$\boldsymbol{f}(k)$是均衡器权向量；$z(k)$是均衡器输出信号，最接近信源信号 $a(k)$，均衡后的 $z(k)$和 $a(k)$之间的误差非常小，$z(k)$经过判决后就

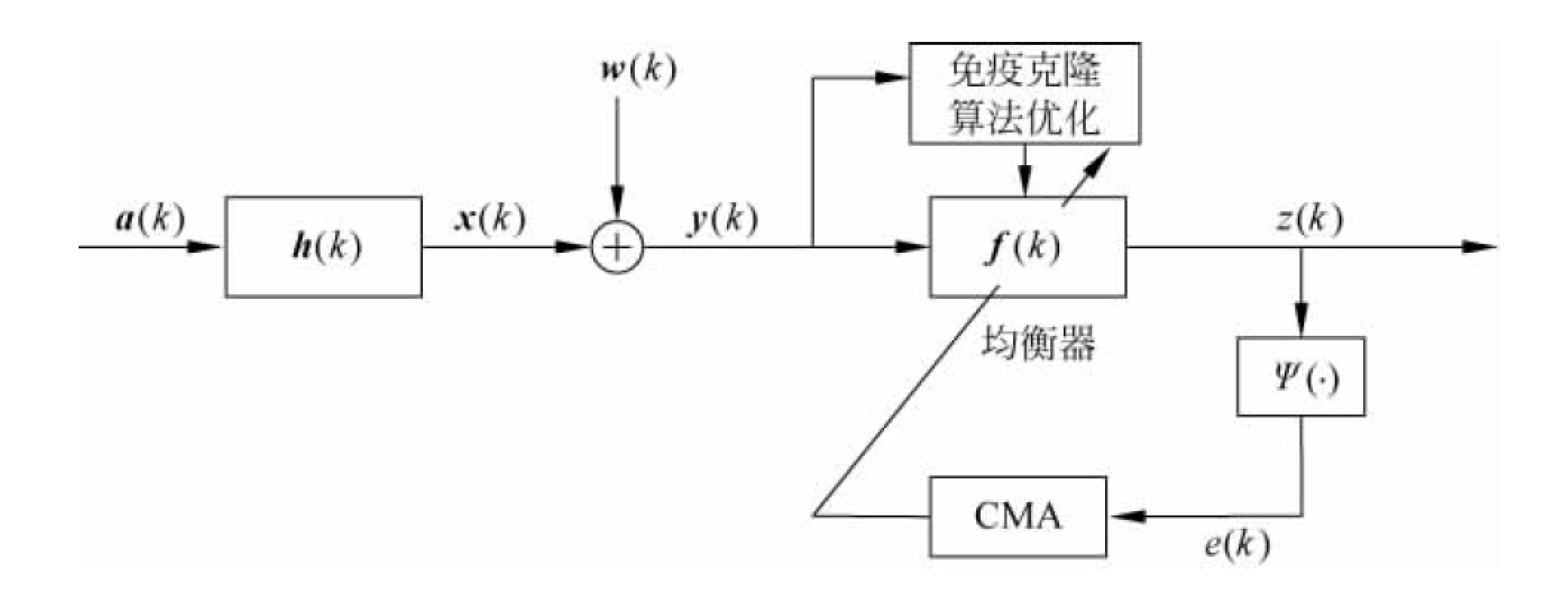



图 4.7 CAS-CMA 原理

能够准确地表示信源信号 $\boldsymbol{a}(k)$；$\Psi(\cdot)$是误差生成函数，$e(k)$是误差函数。

$$y(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+w(k) \tag{4.2.1}$$

$$z(k)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k)=\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{4.2.2}$$

CMA 误差函数为

$$e(k)=|z(k)|^2-R^2 \tag{4.2.3}$$

式中，$R^2$ 为 CMA 模值。

CMA 权向量迭代公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-2\mu e(k)z(k)\boldsymbol{y}^*(k) \tag{4.2.4}$$

免疫克隆优化均衡器权向量 $\boldsymbol{f}(k)$的基本思路是：将抗原作为目标函数或称为亲和度函数，通过模拟生物免疫系统对抗体克隆选择、变异的原理，利用抗体克隆扩大搜索范围、变异保持多样性的特点，把均衡器权向量作为免疫克隆选择算法的决策变量，将均衡器的输入信号作为免疫克隆选择算法的输入，并结合 CMA 的代价函数，确定免疫克隆选择算法优化的目标函数即亲和度函数，利用免疫克隆选择算法来求解均衡器代价函数，快速搜索寻找到均衡器最优权向量，其流程如图 4.8 所示。

设接收信号序列的长度为 $N$，由均衡器误差的时间平均来表示均衡器的代价函数，其代价函数表示为[12]

$$J_{\mathrm{CMA}}(k)=\sum_{k=1}^{K}(|z(k)|^2-R^2)^2/N \tag{4.2.5}$$

这里，首先利用 CMA 来实现均衡，然后利用免疫克隆算法进化的每代

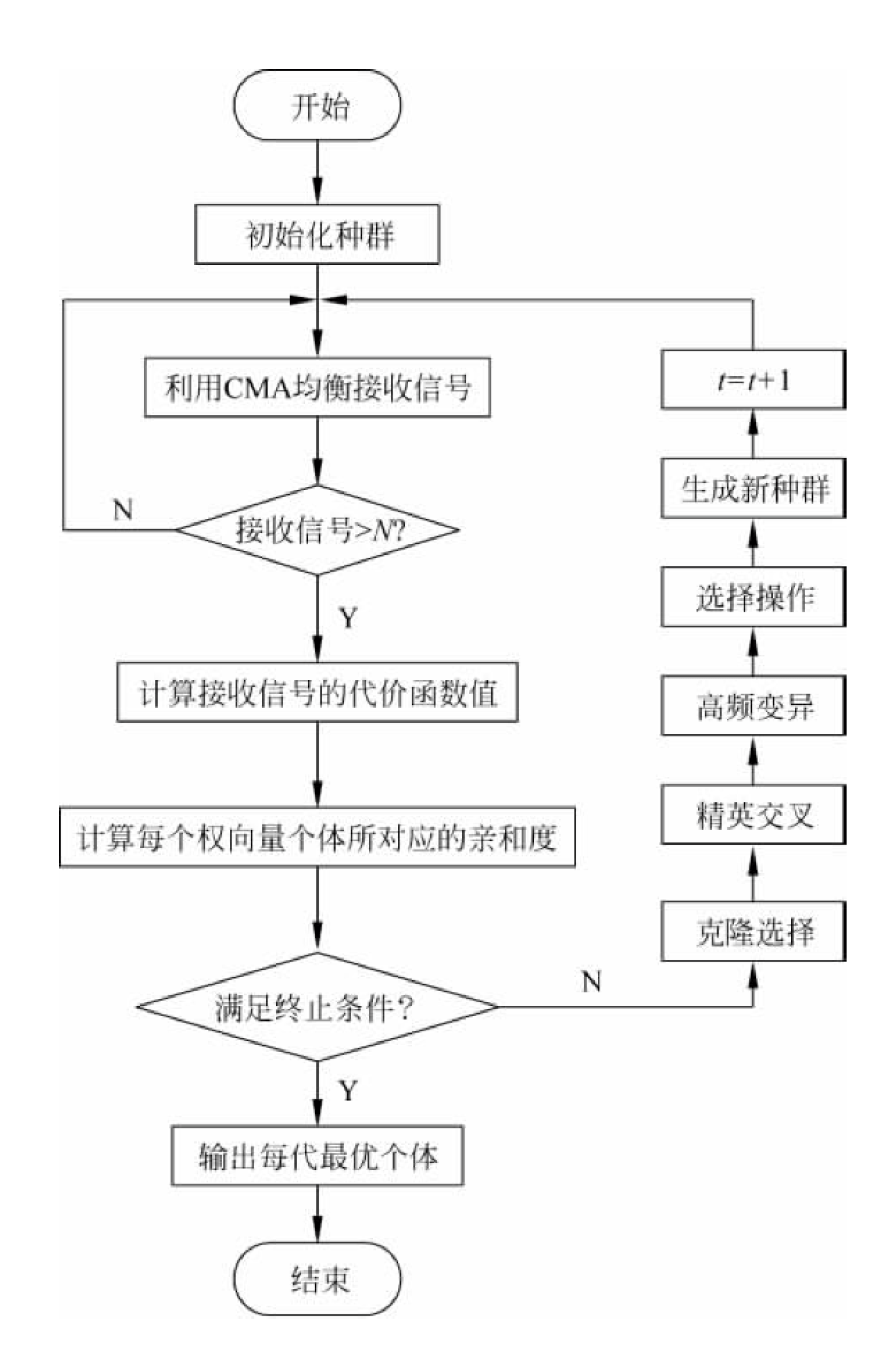



图 4.8　免疫克隆优化 CMA 流程

都依次接收 $N$ 个输入信号进行进化操作，并将进化所产生的新的种群作为下一代进化的初始种群，其具体的实现步骤如下。

步骤 1：种群初始化。随机产生一定数目的抗体种群，其中的每个抗体分别对应一个均衡器的权向量。

步骤 2：计算亲和度值。结合均衡器的代价函数，将亲和度函数即免疫算法寻优的目标函数定义为

$$\mathrm{Fit}(\boldsymbol{f}) = \frac{1}{1 + J_{\mathrm{CMA}}(\boldsymbol{f})} \tag{4.2.6}$$

式中，$\text{Fit}(f)$的最大值对应 $J_{\text{CMA}}(f)$的最小值，即将盲均衡问题转化为求解最高的亲和度最大值对应的均衡器系数问题。

步骤 3：克隆选择。克隆选择操作是克隆增殖操作的逆操作。该操作是从抗体各自克隆增殖后的子代中选择优秀的个体，从而形成新的抗体群，是一个无性选择过程。一个抗体经过克隆增殖后形成一个亚抗体群，再经过亲和度值成熟操作后通过克隆选择操作实现局部的亲和度值升高。首先对步骤 2 所述抗体群中的抗体按亲和度值从小到大的顺序排列，根据亲和度值大小评价，选择最佳抗体进行克隆扩增操作，得到扩增后的抗体群 $C$，克隆数与亲和度成正比。

步骤 4：精英交叉策略。精英交叉的原理如下：在免疫算法中，首先给定一个精英交叉的概率 $p_{\text{kc}}$，对于克隆抗体群中第 $t$ 代每个个体 $a(t)$产生一个$[0,1]$之间的随机数 $R$，如果 $R$ 小于精英交叉概率 $p_{\text{kc}}$，则 $a(t)$被选中与保存的当前代精英个体 $b(t)$进行交叉，其方法是：将 $a(t)$和 $b(t)$放入一个小的交配池中，根据选定的交叉策略（单点、两点、多点和一致交叉等），对$a(t)$和 $b(t)$进行交叉操作，得到一对子代个体 $a'(t)$和 $b'(t)$。然后，用 $a'(t)$替代种群中的 $a(t)$，$b'(t)$则丢弃不用。

通过精英交叉，抗体群继承了精英个体的优良模式，且不会破坏其优良模式，比传统的交叉操作具有更大优越性，因为精英交叉总是能增加群体中的优良模式，而且精英交叉策略对免疫算法的运行时间、平均收敛代数、标准方差等性能有着极大的改善。

步骤 5：高频变异。将精英交叉后的抗体群 $C$ 中每个克隆抗体进行高频变异，得到变异群 $C^*$。高频变异方法为

$$X = x + \alpha \cdot N(0,1) \tag{4.2.7}$$

$$\alpha = (1/\gamma) \cdot \exp(-\text{Fit}) \tag{4.2.8}$$

式中，$X$ 是变异体；$N(0,1)$是均值为 0、标准方差为 1 的正态随机变量；$\alpha$ 是变异概率系数；$\gamma$ 是变异的控制系数；Fit 是抗体与抗原的亲和度值。高频变异作为克隆选择的主要操作算子，可以防止进化早熟并增加抗体的多样性。

步骤 6：计算亲和度值。重新计算高频变异后的各抗体所对应的亲和

度值。

步骤 7：选择。若克隆变异后的抗体亲和度值最大的抗体比父代抗体的亲和度值大，就从变异群 $C^*$ 中选择 $n$ 个亲和度值大的抗体替换初始抗体群中 $n$ 个亲和度值小的抗体，以保持群体多样性，$n$ 反比于抗体群的平均亲和度值。

步骤 8：判断终止与否。根据抗体的进化代数进行判断，当进化代数小于最大进化代数，则转至步骤 2，重复步骤 2～步骤 5 的操作，直至进化代数大于最大进化代数，如达到终止条件，则程序结束，输出全局最优解。

步骤 9：选择最优权向量个体。考虑到免疫克隆选择算法在抽取最优抗体的实时性和盲均衡算法需要满足迫零条件，求取使目标函数最优时所对应的权向量值，并且将这个权向量作为 CSACMA 的初始化权向量。

### 4.2.2 算法仿真

为了检验 CSA-CMA 的有效性，以 CMA 为比较对象，进行仿真实验。仿真试验中，抗体规模为 100，克隆控制因子为 0.6，精英交叉概率为 0.2，变异概率为 0.1，算法最大迭代次数为 500。

**【实验 4.1】** 当发射信号为 8PSK 时，信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$，信噪比为 20dB，均衡器长度为 16；步长 $\mu_{\text{CMA}}=0.0009$，第 4 个抽头初始化设置为 1，其余为 0；步长 $\mu_{\text{CSA-CMA}}=0.005$。

当发射信号为 16QAM 时，信道 $\boldsymbol{h}=[-0.35\quad 0\quad 0\quad 1]$；信噪比为 20dB，均衡器权长是 16；步长 $\mu_{\text{CMA}}=0.000\,008$，第 6 个抽头初始化设置为 1，其余为 0。步长 $\mu_{\text{CSA-CMA}}=0.000\,03$。

用剩余码间干扰来比较各算法的性能，在眼图完全清晰睁开的前提下，20 次蒙特卡罗仿真结果如图 4.9 所示。

图 4.9(a)表明，在收敛速度上，CSA-CMA 比 CMA 快约 4000 步；同时，在剩余码间干扰上，CSA-CMA 比 CMA 降低了约 5dB。图 4.9(b)表明，在收敛速度上，CSA-CMA 比 CMA 快约 2000 步；同时，在剩余码间干扰上，CSA-CMA 比 CMA 降低了约 1.5dB。

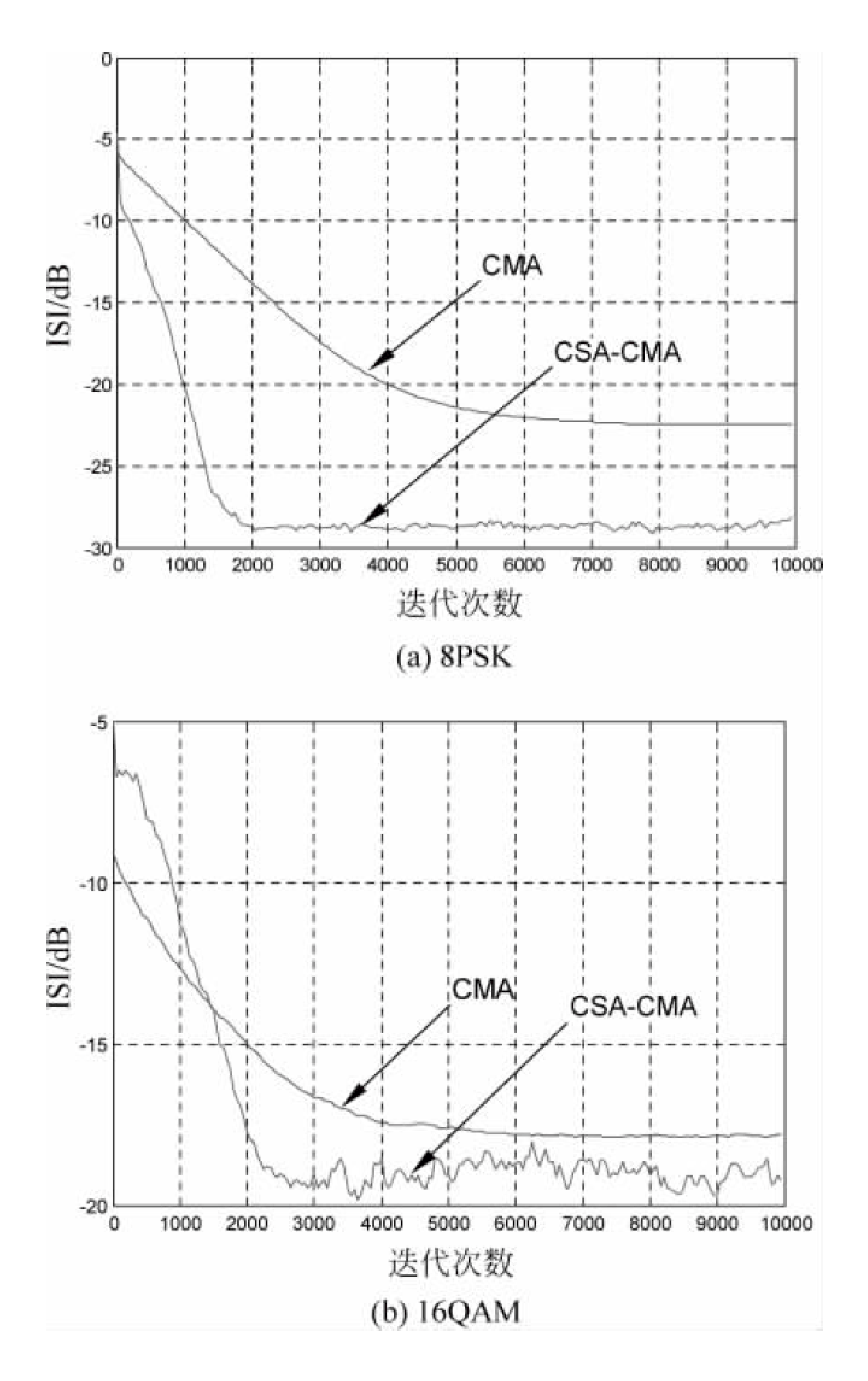



图 4.9　仿真结果

## 4.3　基于免疫克隆算法优化的正交小波支持向机盲均衡算法

由前面讨论知：盲均衡技术是有效补偿信道失真、消除码间干扰、节省大量带宽的有效方法；由正交小波变换和能量归一化对均衡器的输入信号进行处理，可以有效减少各分量之间的自相关性，提高收敛速度；正交小波盲均衡算法因采用梯度下降法对最优权向量搜索，对权向量的初始化同样敏感，易陷入局部极小值，甚至发散。而免疫克隆选择算法是一种较新的全

局搜索算法，可以快速寻找到均衡器权向量的全局最优解，有效防止进化早熟和搜索陷入局部极小值的问题。支持向量机(support vector machine, SVM)是目前解决小样本学习问题的最好机器学习方法，它是一种基于结构风险最小化原则的分类器，利用 SVM 来构造均衡器结构，可以有效解决 CMA 难以满足小样本学习和存在局部收敛的缺陷，但 SVM 的参数设置对最终分类精确度有较大影响[13,14]。因此，将免疫克隆选择算法分别引入到正交小波常模盲均衡算法、正交小波支持向量机常模盲均衡算法中，就得到了基于免疫克隆算法优化的正交小波常模盲均衡算法和基于免疫克隆算法优化的正交小波支持向量机常模盲均衡算法[15-18]。

### 4.3.1　基于免疫克隆算法优化的正交小波常模盲均衡算法

#### 1. 算法原理

均衡器输入信号的自相关矩阵是影响盲均衡算法收敛速度的主要因素，因此，将携带噪声的均衡器输入信号经过正交小波变换后，来调整均衡器权向量，使其自相关性下降，以提高收敛速度；结合免疫克隆算法的全局搜索能力，搜索均衡器最优权向量，可以有效克服易陷入局部收敛的缺点，这样得到的基于免疫克隆算法优化的正交小波常模盲均衡算法(CSA based WTCMA, CSA-WTCMA)，如图 4.10 所示。

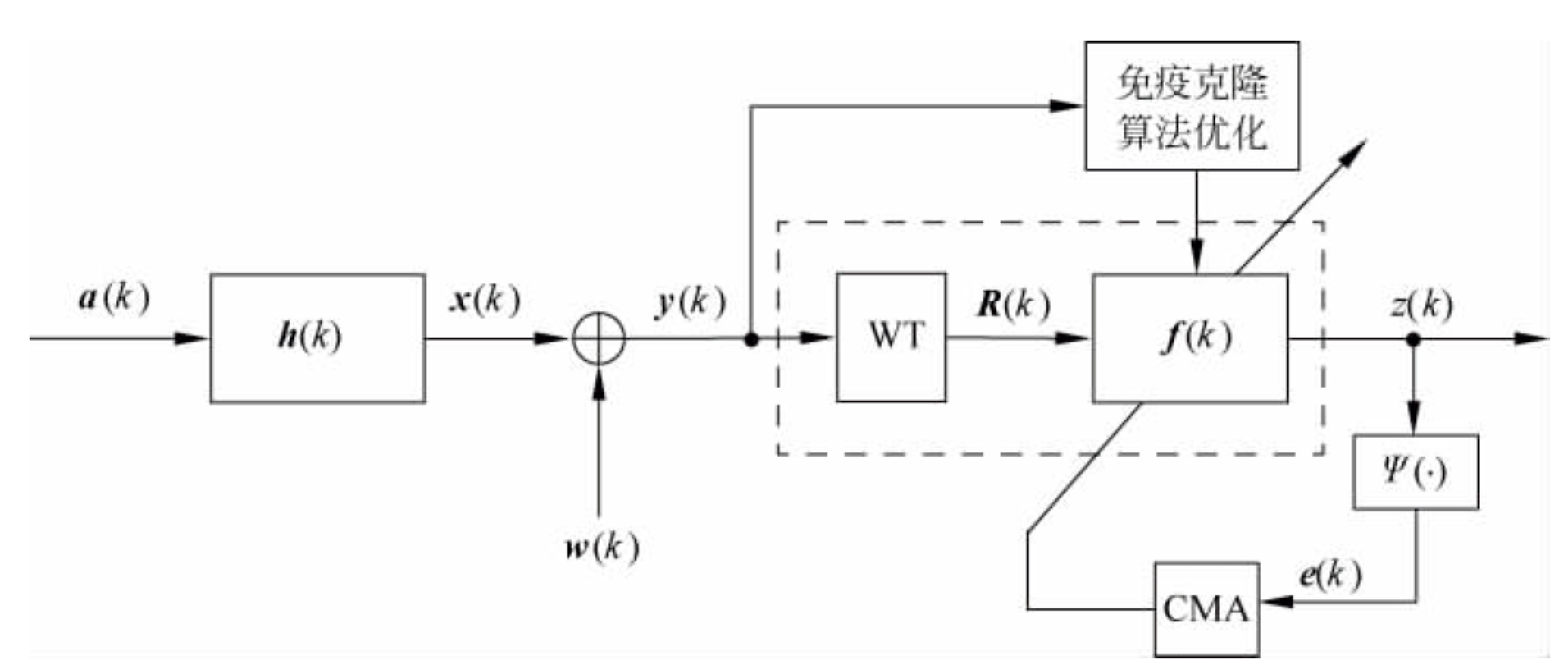



图 4.10　CSA-WTCMA 原理

图 4.10 中，$\boldsymbol{a}(k)$表示发射信号；$\boldsymbol{h}(k)$表示信道脉冲响应向量；$\boldsymbol{w}(k)$表示加性高斯白噪声；$\boldsymbol{y}(k)$表示均衡器输入信号；$\boldsymbol{f}(k)$表示均衡器权向量；$z(k)$表示均衡器输出信号；$\Psi(\cdot)$表示误差生成函数，$e(k)$是误差函数。

正交小波变换函数或免疫克隆算法的输入为

$$y(k) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k) + w(k) \tag{4.3.1}$$

正交小波变换函数的输出或均衡器的输入为

$$\boldsymbol{R}(k) = \boldsymbol{V}\boldsymbol{y}(k) \tag{4.3.2}$$

均衡器的输出为

$$z(k) = \boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{R}(k) \tag{4.3.3}$$

均衡器的误差为

$$e(k) = R^2 - |z(k)|^2 \tag{4.3.4}$$

均衡器权向量的迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu\hat{\boldsymbol{R}}^{-1}(k)e(k)\boldsymbol{R}^*(k) \tag{4.3.5}$$

式中，$\hat{\boldsymbol{R}}(k)=\mathrm{diag}[\sigma_{j,0}^2(k),\sigma_{j,1}^2(k),\cdots,\sigma_{J,k_J}^2(k),\sigma_{J+1,0}^2(k),\cdots,\sigma_{J+1,k_J}^2(k)]$，$\mu$ 为迭代步长，$\sigma_{j,n}^2(k)$、$\sigma_{J+1,n}^2(k)$分别表示对小波变换系数 $u_{j,n}(k)$、尺度变换系数 $s_{J,n}(k)$的平均功率估计，其递推公式为

$$\hat{\sigma}_{j,n}^2(k+1) = \beta\hat{\sigma}_{j,n}^2(k) + (1-\beta)\,|u_{j,n}(k)|^2 \tag{4.3.6}$$

$$\hat{\sigma}_{J+1,n}^2(k+1) = \beta\hat{\sigma}_{J+1,n}^2(k) + (1-\beta)\,|s_{J,n}(k)|^2 \tag{4.3.7}$$

式中，$\beta$ 为平滑因子，且 $0<\beta<1$，一般取 $\beta$ 值比较接近于 1。以上就是 WTCMA。免疫克隆算法优化 WTCMA 流程如图 4.11 所示。这时就得到 CSA-WTCMA。

## 2. 算法仿真

为了验证 CSA-WTCMA 的有效性，以 WTCMA 作为比较对象，进行仿真实验。

**【实验 4.2】** 当发射信号为 8PSK 时，信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；信噪比为 20dB，均衡器权长均为 16。步长 $\mu_{\mathrm{WTCMA}}=$

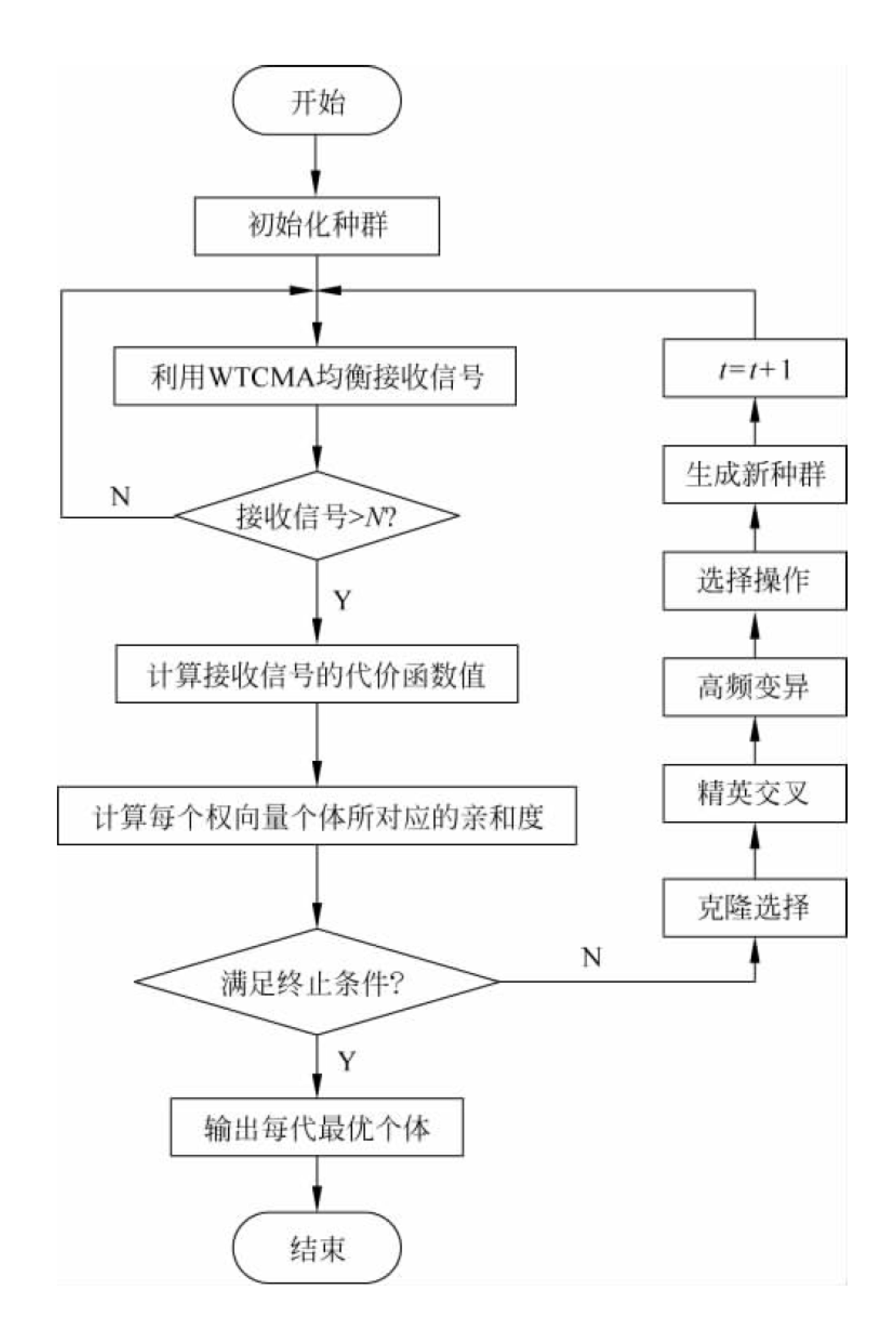



图 4.11 免疫克隆算法优化 WTCMA 流程

0.001,第 10 个抽头初始化设置为 1,其余为 0;步长 $\mu_{\text{CSA-WTCMA}}=0.005$。对每个信道的输入信号采用 DB4 正交小波进行分解,分解层次是 2,功率初始值设置为 4,遗忘因子 $\beta=0.9999$。

当发射信号为 16QAM 时,信道 $\boldsymbol{h}=[0.9656 \quad -0.0906 \quad 0.0578 \quad 0.2368]$;信噪比为 20dB,均衡器权长均为 16。步长 $\mu_{\text{WTCMA}}=0.000\,55$,第 10 个抽头初始化设置为 1,其余为 0;步长 $\mu_{\text{CSA-WTCMA}}=0.000\,18$。对每个信道的输入信号采用 DB4 正交小波进行分解,分解层次是 2,功率初始值设置

为 4,遗忘因子 $\beta=0.95$。

在眼图充分睁开的条件下,500 次蒙特卡罗仿真结果如图 4.12 所示。

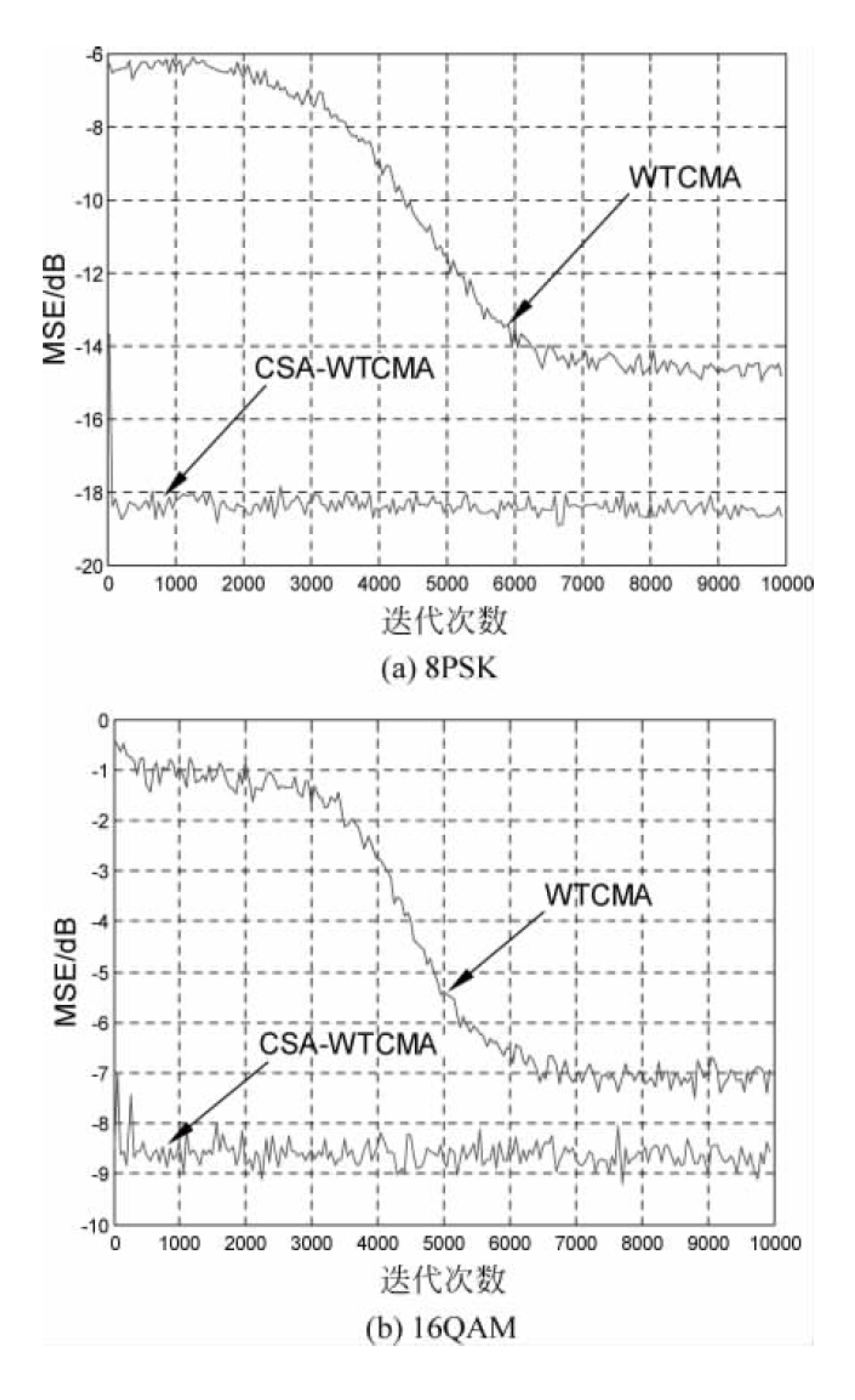



图 4.12 均方误差收敛曲线

图 4.12(a)表明,在收敛速度上,CSA-WTCMA 比 WTCMA 快了约 6000 步。在稳态误差上,CSA-WTCMA 比 WTCMA 减小了近 4dB。图 4.12(b)表明,在收敛速度上,CSA-WTCMA 比 WTCMA 快了约 6500 步,在稳态误差上,CSA-WTCMA 比 WTCMA 减小了近 1.5dB。

### 4.3.2　基于免疫克隆选择算法优化的正交小波支持向量机常模盲均衡算法

#### 1. 基于支持向量机的常模盲均衡算法

基于支持向量机的常模盲均衡系统,如图 4.13 所示。

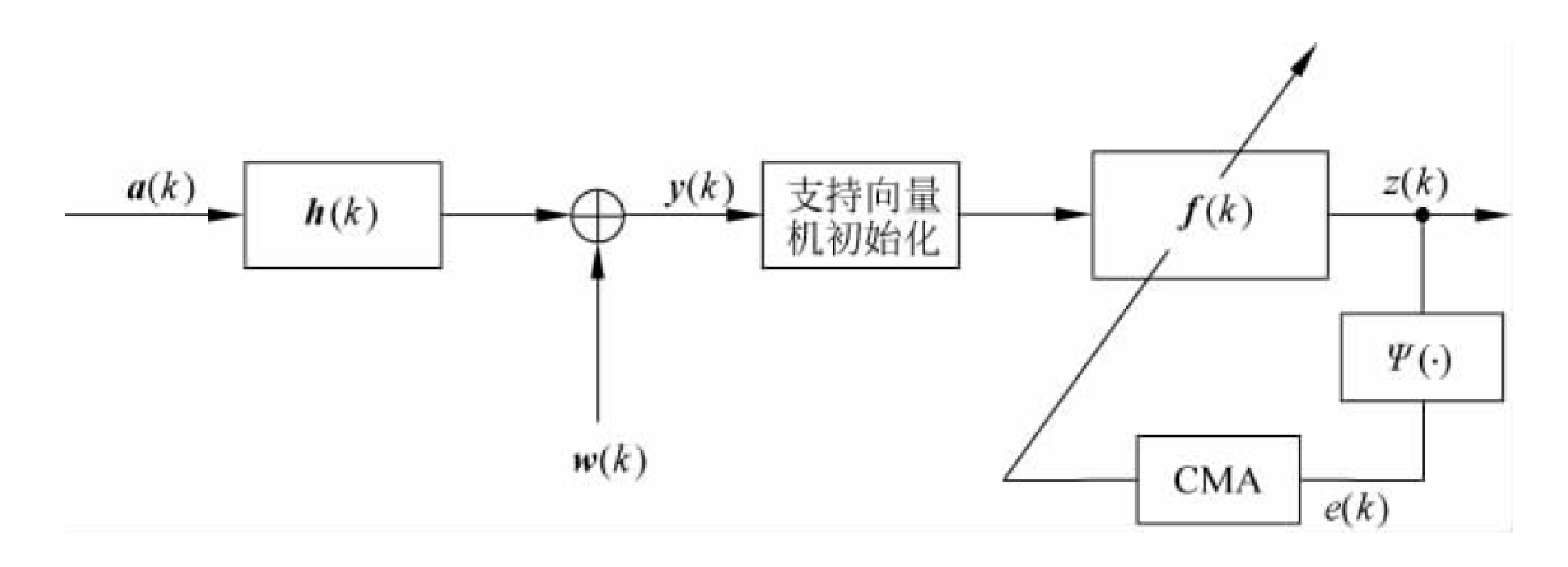



图 4.13　基于支持向量机的常模盲均衡系统

根据码间干扰(inter-symbol interference,ISI)空间几何理论,将盲均衡问题作为支持向量机(support vector machine,SVM)所构造的函数回归拟合问题,函数的参数为均衡器权向量。基于 SVM 的盲均衡器算法结构,如图 4.14 所示。

图 4.14 中,均衡器接收 $N$ 组信号的向量为 $\boldsymbol{y}_n(n=1,2,\cdots,N)$,其中 $\boldsymbol{y}_n=\{y_n(k),y_n(k+1),\cdots,y_n(k-M+1)\}$,$M$ 为均衡器的长度,$\boldsymbol{f}(k)$为均衡器权向量。令 $a(k)$是以 $T$ 为周期的发射信号序列,则信道输出即接收的观测序列为

$$y(k)=\sum_{n=1}^{M-1}a(n)h(k-n)+w(k) \tag{4.3.8}$$

均衡器输出为

$$Z(k)=\sum_{n=0}^{M-1}f(n)y(k-n)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{y}(k) \tag{4.3.9}$$

对常数模信号,根据结构风险最小化原则(SRM),以精度 $\varepsilon$ 估计均衡器权向量 $\boldsymbol{f}(k)$,其最小化代价函数为

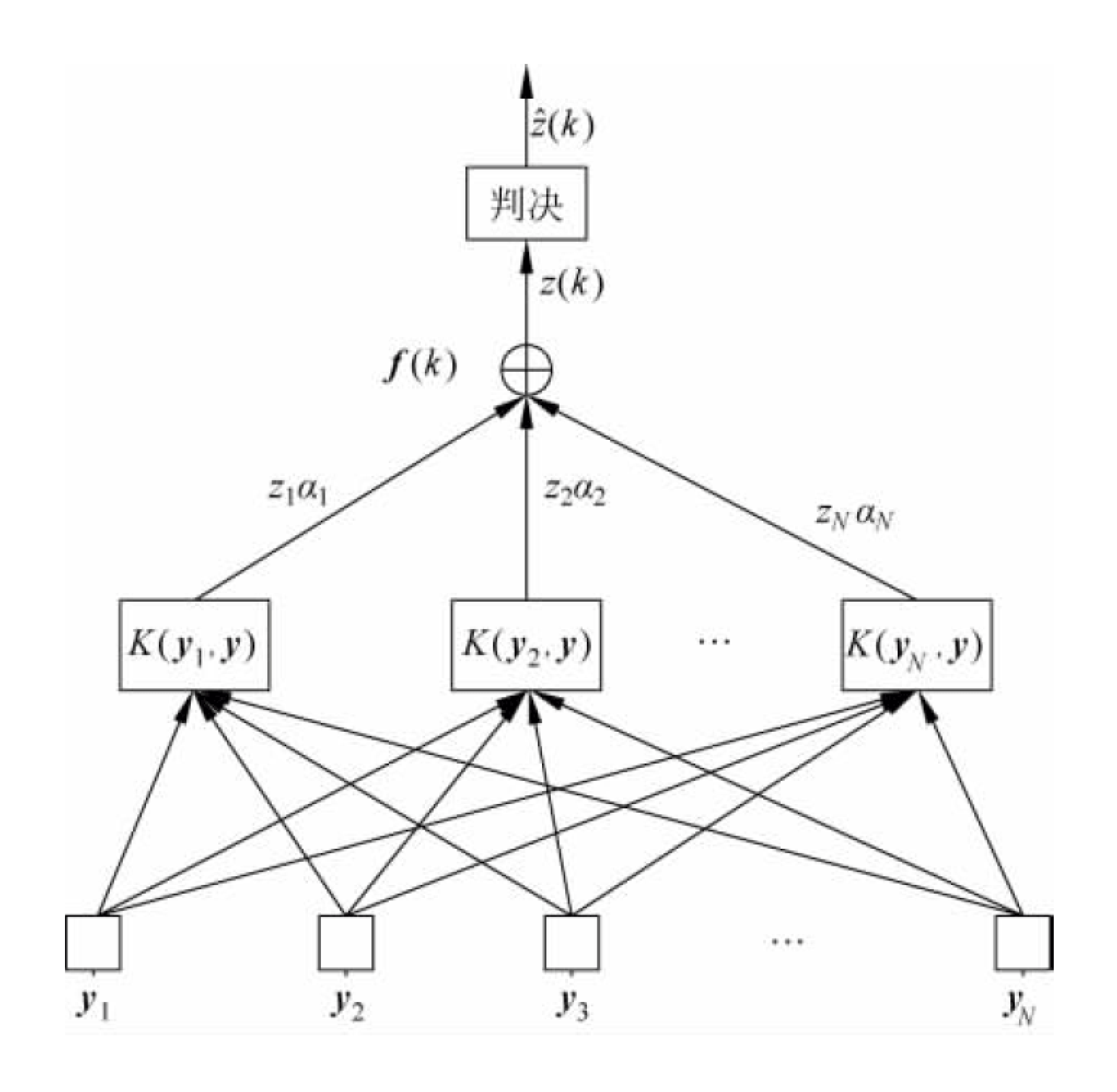



图 4.14 基于 SVM 的盲均衡器结构

$$J(\boldsymbol{f}(k))=\frac{1}{2}\|\boldsymbol{f}(k)\|^2+C\sum_{n=1}^{N}|R^2-([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)^2|_\varepsilon \tag{4.3.10}$$

式中,$C$ 为惩罚系数,由 Vapnik 的 $\varepsilon$ 不敏感损失函数,得

$$|R^2-([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)^2|_\varepsilon=\{0,(|R^2-[\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)^2|-\varepsilon\} \tag{4.3.11}$$

引入松弛变量 $\xi(n)$ 和 $\tilde{\xi}(n)$,最小化式(4.3.10)可等价为问题

$$\min\frac{1}{2}\|\boldsymbol{f}(k)\|^2+C\sum_{n=1}^{N}(\xi(n)+\tilde{\xi}(n)) \tag{4.3.12}$$

约束条件为

$$\begin{cases}R^2-([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)^2\leqslant\varepsilon+\xi(n)\\([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)^2-R^2\leqslant\varepsilon+\tilde{\xi}(n)\\\xi(n),\qquad\tilde{\xi}(n)\geqslant 0\end{cases} \tag{4.3.13}$$

由于约束条件中对于均衡器权向量 $\boldsymbol{f}(k)$ 含有二次项,上面的最优化问题无法通过 SVM 所采用的线性规划方法求解。根据文献[13]中迭代权二次规划算法(iterative reweighted quadratic programming,IRWQP)来解决

这一问题，可以将式(4.3.12)中的二次约束改写为线性约束，即

$$\begin{cases}([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)Z(n)-R^2\leqslant\varepsilon+\tilde{\xi}(n)\\ R^2-([\boldsymbol{f}(k)]^{\mathrm{T}}\boldsymbol{y}_n)Z(n)\leqslant\varepsilon+\xi(n)\end{cases}\tag{4.3.14}$$

为使原始问题的对偶问题导出，引入 Lagrange 函数

$$\begin{aligned}L(\boldsymbol{f},\xi,\tilde{\xi},\alpha,\tilde{\alpha},b,\tilde{b})=&\frac{1}{2}\|\boldsymbol{f}(k)\|^2+C\sum_{n=1}^{N}[\xi(n)+\tilde{\xi}(n)]-\\&\sum_{n=1}^{N}(b_n\xi(n)+\tilde{b}_n\tilde{\xi}(n))-\\&\sum_{n=1}^{N}\alpha(n)[R^2-([\boldsymbol{f}(k)]^2\boldsymbol{y}_n)Z(n)+\varepsilon+\xi(n)]-\\&\sum_{n=1}^{N}\tilde{\alpha}(n)[([\boldsymbol{f}(k)]^2\boldsymbol{y}_n)Z(n)-R^2+\varepsilon+\tilde{\xi}(n)]\end{aligned}\tag{4.3.15}$$

将式(4.3.9)～式(4.3.15)的原始最优化问题转换为凸二次规划问题(对偶问题)，即

$$\begin{aligned}\max&-\frac{1}{2}\sum_{n,j=1}^{N}(\tilde{\alpha}(j)-\alpha(j))(\tilde{\alpha}(n)-\alpha(n))(Z(j)Z(n))K\langle\boldsymbol{y}_j,\boldsymbol{y}_n\rangle\\&-\varepsilon\sum_{j=1}^{N}(\tilde{\alpha}(j)+\alpha(j))+\sum_{j=1}^{N}(\tilde{\alpha}(j)-\alpha(j))\end{aligned}\tag{4.3.16}$$

约束条件为

$$\begin{cases}\displaystyle\sum_{j=1}^{N}(\tilde{\alpha}(j)-\alpha(j))=0\\ 0\leqslant\tilde{\alpha}(j)\leqslant C,j=1,2,\cdots,N\end{cases}\tag{4.3.17}$$

通过比较原始问题与对偶问题，均衡器权向量为

$$\boldsymbol{f}(k)=\sum_{n=1}^{k}(\tilde{\alpha}(n)-\alpha(n))Z(n)\boldsymbol{y}_n\tag{4.3.18}$$

式中，Lagrange 乘子 $\alpha(n)$ 和 $\tilde{\alpha}(n)$ 可以通过式(4.3.16)和式(4.3.17)求得。

均衡器权向量的更新公式为

$$\boldsymbol{f}(k)=\lambda\boldsymbol{f}(k-1)+(1-\lambda)\boldsymbol{f}(k-2)\tag{4.3.19}$$

式中，$k$ 表示迭代次数，$\lambda$ 表示迭代步长。

2. 基于正交小波变换的支持向量机常模盲均衡算法

基于正交小波变换的支持向量机盲均衡算法(WTCMA based on SVM,WTCMA-SVM),如图 4.15 所示。

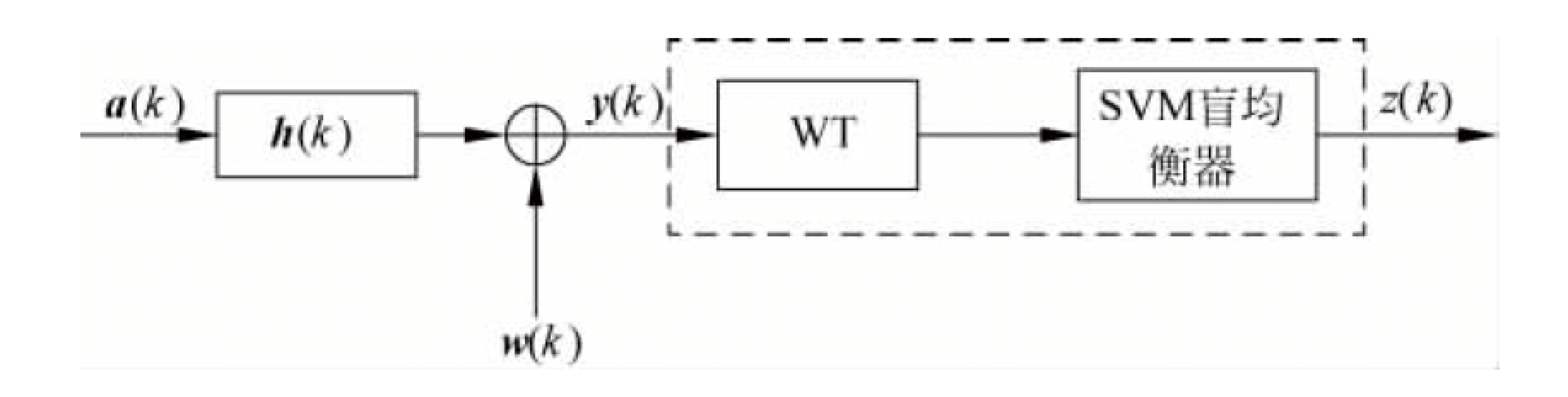



图 4.15 WTCMA-SVM 原理

均衡器输出为

$$Z(k) = \boldsymbol{f}^{\mathrm{H}}(k)\boldsymbol{R}(k) \tag{4.3.20}$$

均衡器权向量为

$$\boldsymbol{f}(k) = \sum_{i=1}^{k}(\tilde{\alpha}(i) - \alpha(i))[\hat{\boldsymbol{R}}(i)]^{-1}Z(i)\boldsymbol{R}(i) \tag{4.3.21}$$

式中,$[\hat{\boldsymbol{R}}(i)]^{-1} = \mathrm{diag}\{\sigma_{1,0}^2(i), \sigma_{1,1}^2(i), \cdots, \sigma_{J,k_j-1}^2(i), \cdots, \sigma_{J+1,k_j-1}^2(i)\}$为能量归一化矩阵,并且

$$\sigma_{J,k}^2(i+1) = \beta\sigma_{J,k}^2(i) + (1-\beta)\mid u_{j,k}(i)\mid^2 \tag{4.3.22}$$

$$\sigma_{J+1,k}^2(i+1) = \beta\sigma_{J+1,k}^2(i) + (1-\beta)\mid s_{j,k}(i)\mid^2 \tag{4.3.23}$$

式中,$\beta$表示遗忘因子。先经过对信号进行正交小波变换并作归一化处理,再对权向量 $\boldsymbol{f}(k)$按照式(4.3.21)进行循环迭代便构成了基于正交小波变换的支持向量机常模盲均衡算法。

3. SVM 中参数选择方法

由于 SVM 参数选择问题可以归结为一个多参数优化问题,SVM 模型中主要包括两类参数:一类是基本参数,例如,标准支持向量机的均衡参数 $C$、$\nu$-SVR 中的 $\nu$ 和$\varepsilon$-SVR 中的 $\varepsilon$ 等;另一类是核函数和核函数中的相关参数,例如,多项式核函数中的常数 $c$、指数 $d$,高斯核函数中的核宽度 $\sigma$ 等。

支持向量机与其他学习算法一样，性能都依赖于学习机的参数选择。但到目前为止，支持向量机参数选择还没有公认的最好方法。随着智能优化算法的不断改进和完善，通过给定各参数的区间，将不同优化算法用到 SVM 的参数选择问题上，优化参数空间内参数寻优过程，最终得到最优的模型参数，下面介绍几种常见的 SVM 参数选择算法[19~20]。

(1) 基于遗传算法的 SVM 参数选择[21]

将 SVM 参数选择问题看作一个优化模型，所求解为惩罚因子 $C$ 和核函数的相关参数，其中目标函数如式(4.3.24)所示。用遗传算法进行求解如下：将训练样本集分为训练子集和测试子集两部分，首先随机产生初始种群并进行编码，用训练子集对初始种群对应的参数值进行训练，并用测试子集对训练好的 SVM 进行测试，将式(4.3.24)所求结果作为遗传算法中的适应度函数；判定是否满足算法终止条件，如不满足，继续执行遗传操作产生新一代种群，否则，得到最优解，即为最佳参数。

$$\mathrm{RA}=\frac{\text{测试数据集中正确类别的样本数目}}{\text{测试数据集样本个数}} \tag{4.3.24}$$

(2) 基于离散微粒群算法的 SVM 参数选择[22]

如果一组训练样本能够被一个最优分类面和支持向量机分开，则将训练样本中平均的支持向量占总训练样本数的比例作为测试样本分类错误率期望的上界，即：$E(p(\mathrm{error}))\leqslant E$(支持向量个数)/训练样本－1，选取适应函数 $d_0$＝支持向量个数/训练样本－1，并将微粒的搜索空间确定为需要搜索的参数，利用微粒群算法对 SVM 的参数进行搜索，最终得到最优参数组合。

(3) 基于模拟退火算法的 SVM 参数选择[23]

支持向量机算法的超参数优化问题是一个需要同时优化分类误差 $\mathrm{Fit}_1(x)=\mathrm{Fit}_{\mathrm{error}}$ 和模型复杂性 $\mathrm{Fit}_2(x)=\mathrm{Fit}_{\mathrm{complex}}$ 两个目标函数的多目标优化问题。对于二值分类问题，分类误差用分类精度表示为

$$\mathrm{Fit}_1(x)=\mathrm{Fit}_{\mathrm{error}}=1-\mathrm{Fit}_{\mathrm{accuracy}}=1-N_{\mathrm{coorect}}/N_{\mathrm{sum}}$$

式中，$\mathrm{Fit}_{\mathrm{accuracy}}$ 为分类精度，$N_{\mathrm{sum}}$ 为样本总数，$N_{\mathrm{coorect}}$ 为正确分类的样本数量。

支持向量机寻优参数与模拟退火算法结合，在某一初温下，随着温度参数的不断下降，根据概率突跳特性在解空间中随机搜索目标函数的全局最优解作为最优参数。

上述几种优化支持向量机参数选择算法，虽能够满足一般参数选择的自动性和高效性，但由于这些优化算法本身都存在着容易陷入局部极值的缺陷，所以优化问题模型的构建和优化算法的选取严重影响优化结果。

### 4. 免疫克隆选择算法优化正交小波支持向量机常模盲均衡算法性能

(1) 算法优化原理

在正交小波支持向量机常模盲均衡算法中，需要确定一些参数的取值。例如，核函数、惩罚参数 $C$、$\varepsilon$-不敏感损失函数等，不同参数的设置会严重影响 SVM 机器学习的性能，所以大多都是通过反复的实验，以及人的主观经验来选择需要的参数，并且需要付出较多的时间代价。其中，惩罚函数 $C$ 和 $\varepsilon$-不敏感损失函数的 $\varepsilon$ 宽度是控制逼近函数 VC 维数（逼近函数集合大小的定量指标）的自由参数，由于其在选择时必须同时调整，具有一定的复杂性。

针对这一缺点，本节利用免疫克隆选择算法全局寻优的特性，对支持向量机中的参数进行优化选择，以自动获得最佳模型参数。具体思路是将 SVM 的训练样本设置为抗原，每个抗体分别对应于参数 $C$ 和 $\varepsilon$ 中的一组取值，首先确定参数 $C$ 和 $\varepsilon$ 的取值范围，通过模拟生物免疫系统中抗体克隆扩大搜索范围、变异保持多样性的特点，寻找目标函数的参数最优解，并作为支持向量机中惩罚参数 $C$、$\varepsilon$-不敏感损失函数。将免疫克隆选择算法应用于支持向量机参数优化选择，以改善支持向量机初始化权向量的性能。这样得到的基于免疫克隆选择算法优化的正交小波支持向量机常模盲均衡算法（immune optimization algorithm based WTCMA and SVM，IO-WTCMA-SVM）原理，如图 4.16 所示。其优化流程如图 4.17 所示。

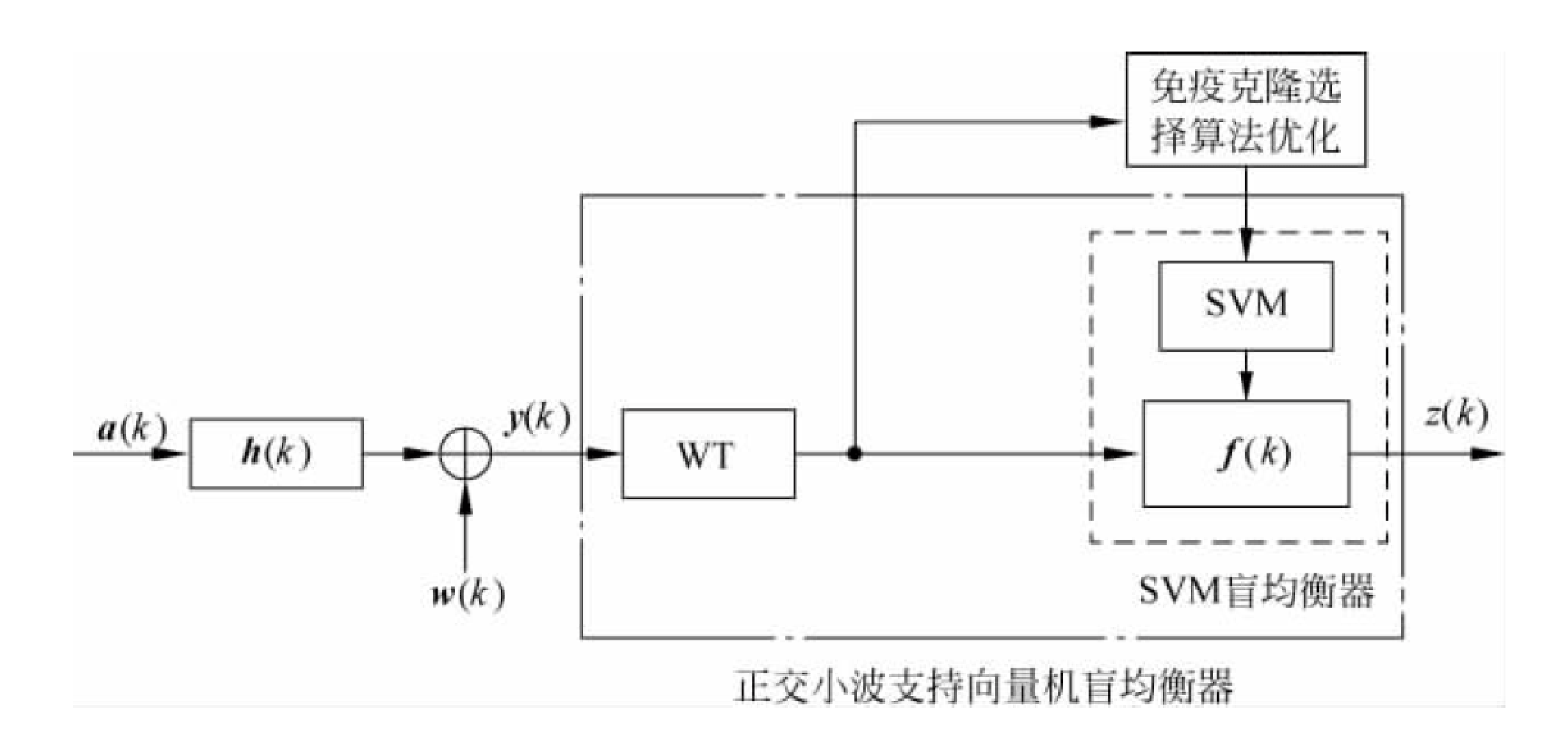



图 4.16　IO-WTCMA-SVM 原理

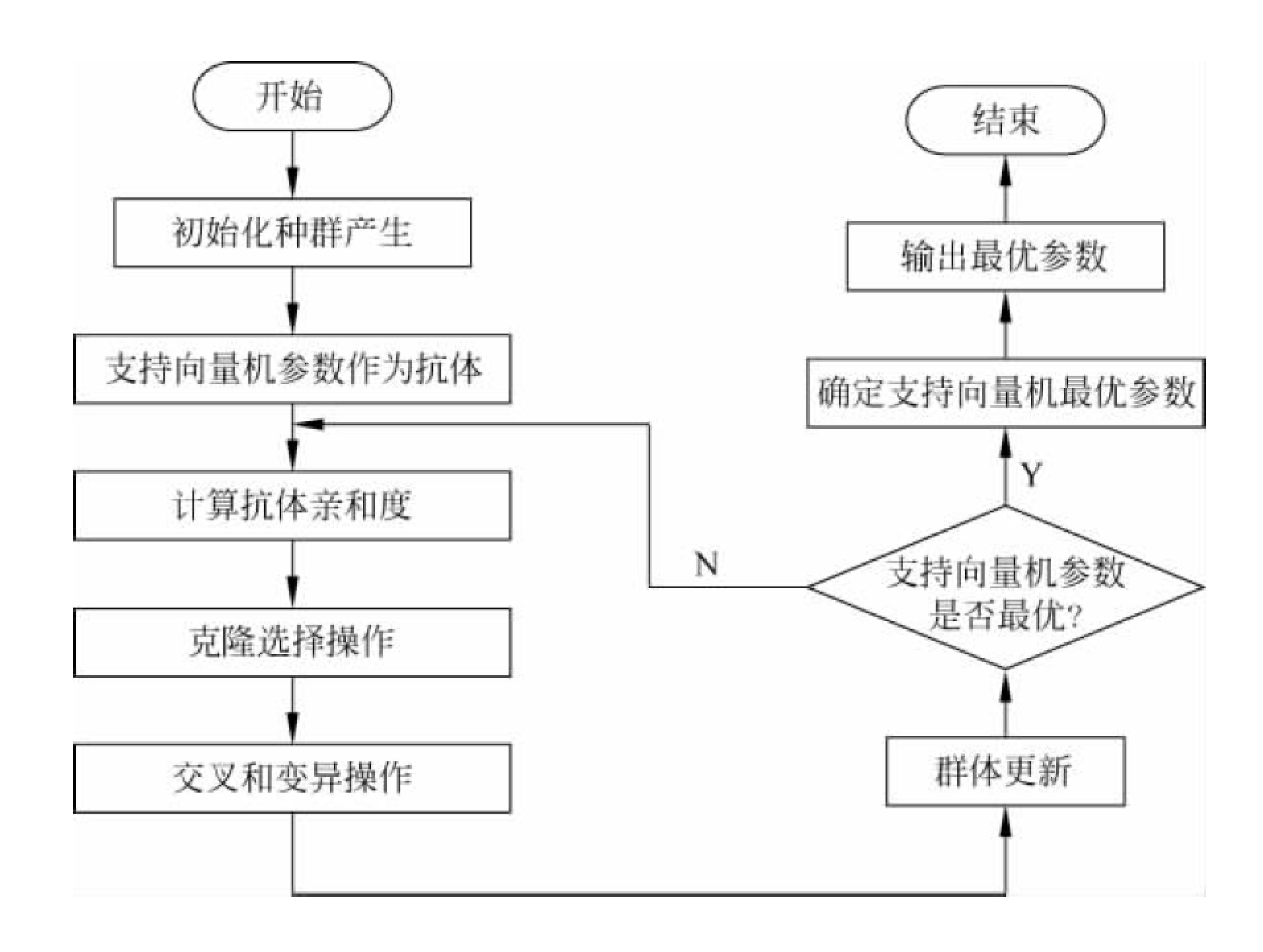



图 4.17　免疫克隆选择算法优化支持向量机参数流程

(2) 算法仿真

为了验证 IO-WTCMA-SVM 的有效性，以 SVM 和 WTCMA-SVM 作为比较对象，进行仿真实验。实验中，抗体规模为 100，克隆控制因子为 0.6，精英交叉概率为 0.2，变异概率为 0.1，算法最大迭代次数为 200。参数 $C$ 和 $\varepsilon$ 优化取值范围设为：$1\leqslant C\leqslant 30$，$0.00001\leqslant\varepsilon\leqslant 0.1$，信噪比为 20dB，均衡器权长均为 16。

**【实验 4.3】** 发射信号为 8PSK，信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；SVM 算法中，迭代步长 $\lambda=0.9$，$N=200$，$C=25$，$\varepsilon=0.3$；WTCMA-SVM 中，迭代步长 $\lambda=0.9$，$N=200$，$C=25$，$\varepsilon=0.3$；IO-WTCMA-SVM 中，$N=200$，迭代步长 $\lambda=0.9$，免疫优化选择最优参数为 $C=6.1211$，$\varepsilon=0.0377$；对输入信号采用 DB2 正交小波进行分解，分解层次是 3，功率初始值设置为 4，遗忘因子 $\beta=0.9999$。仿真结果如图 4.18 所示。

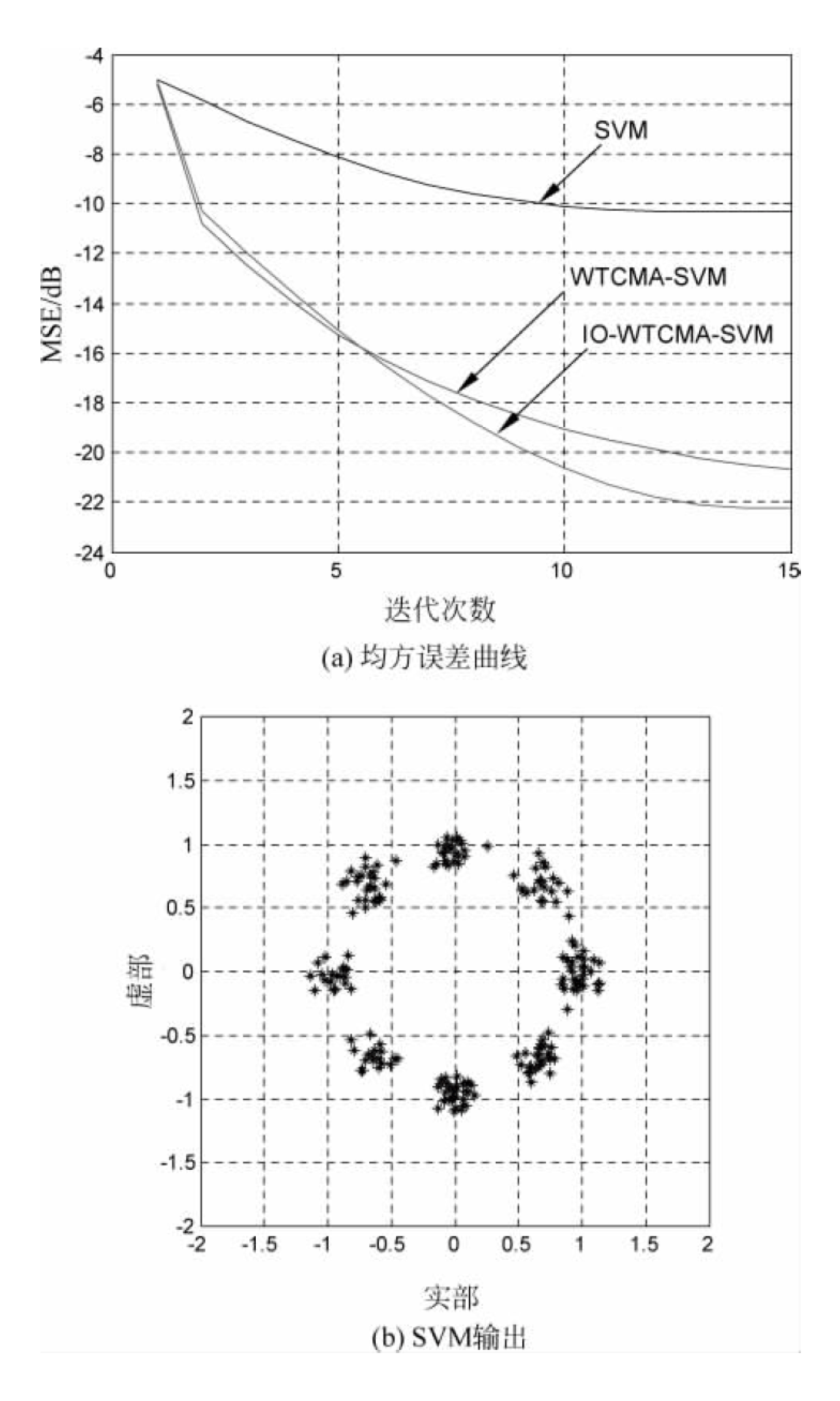
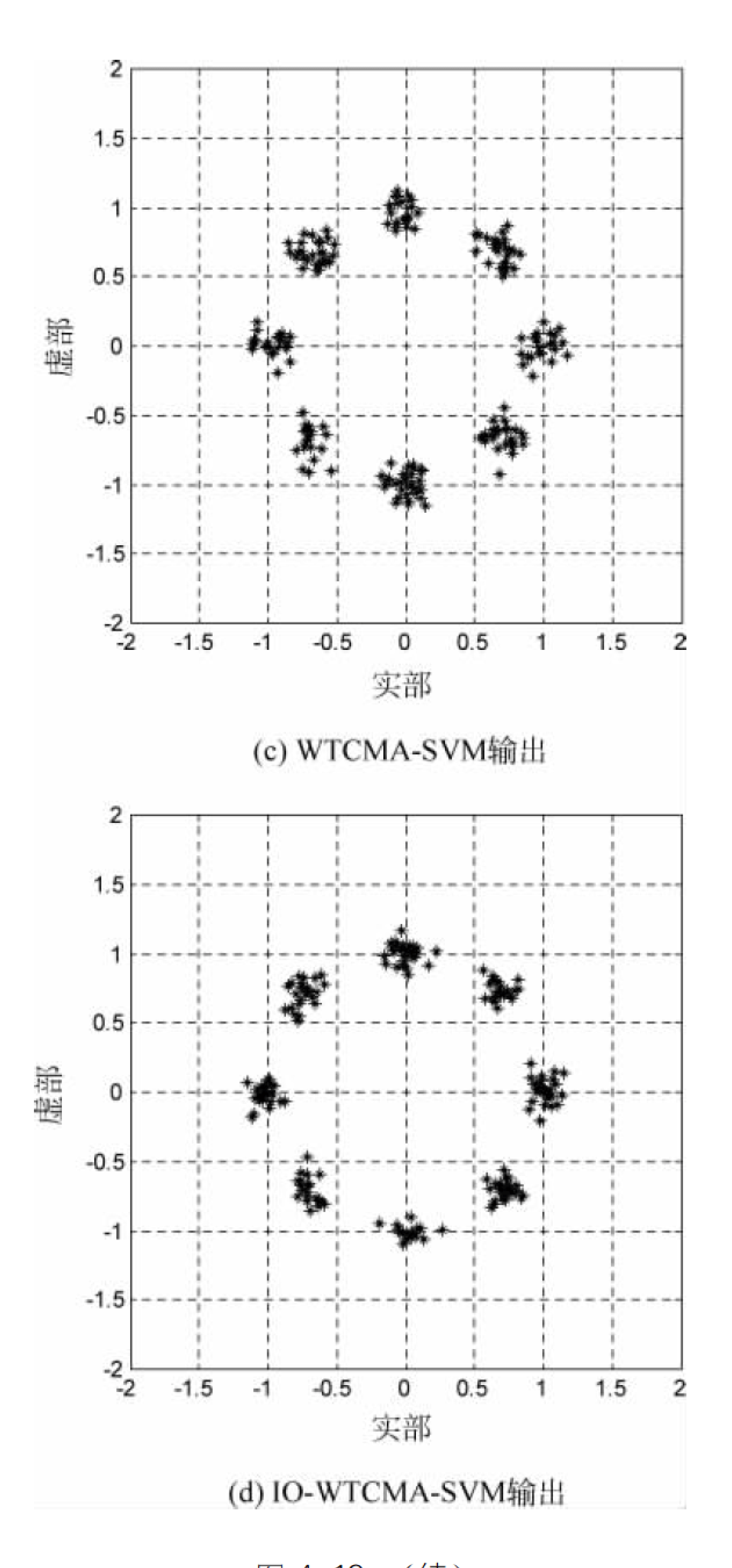



图 4.18　仿真结果



(c) WTCMA-SVM输出

(d) IO-WTCMA-SVM输出

图 4.18 （续）

图 4.18(a)表明，在只有 200 个训练点的情况下，IO-WTCMA-SVM 比 SVM 算法稳态误差减小了 12dB，比 WTCMA-SVM 减小了 2dB，有效消除了 WTCMA-SVM 存在的误收敛。图 4.18(b)～(d)表明，由 IO-WTCMA-SVM 所得到的星座图更为清晰、紧凑，更好地消除了码间干扰。

## 4.4 基于自适应免疫克隆选择算法优化的正交小波超指数迭代常模盲均衡算法

### 4.4.1 自适应免疫克隆选择算法

正交小波超指数迭代盲均衡算法(WTSEI)是利用快速梯度下降搜索法对均衡器权向量进行搜索,而免疫克隆选择算法是一种全局随机搜索的方法,可以弥补 WTSEI 搜索过程中陷于局部收敛的缺陷,但在标准免疫克隆选择算法中,针对不同的优化问题,需要反复试验来确定抗体的克隆数和高频变异概率。而自适应免疫克隆选择算法[24]是一种自适应选择确定抗体的克隆数和高频变异概率,在进化初期,为了保证抗体在大范围内搜索,利用操作算子进行全局进化,有效避免过早收敛;在进化后期,当接近全局最优解时,抗体就跳出局部极值点,在局部范围内搜索进化,能更好提高解的精度。这就要求在进化过程中操作算子需根据抗体的进化情况,随时改变进化的策略。而且,高频变异算子是决定克隆选择算法中收敛性的核心操作。因此,本节给出了一种自适应免疫克隆的正交小波超指数迭代盲均衡算法,根据亲和度值来调节每个抗体的克隆数。每个抗体的克隆数计算为[25]

$$C=\begin{cases}C_{\max}\left\lfloor\dfrac{(\mathrm{Fit}-\mathrm{Fit}_{\min})}{(\mathrm{Fit}_{\mathrm{avg}}-\mathrm{Fit}_{\min})}\right\rfloor, & \mathrm{Fit}>\mathrm{Fit}_{\mathrm{avg}}\\ 0, & \mathrm{Fit}\leqslant\mathrm{Fit}_{\mathrm{avg}}\end{cases}\tag{4.4.1}$$

式中,$C$ 表示需要克隆的抗体克隆数;$C_{\max}$表示所给定的最大克隆控制因子;Fit 表示需要克隆的抗体亲和度值;$\mathrm{Fit}_{\min}$表示抗体群中抗体亲和度最小值;$\mathrm{Fit}_{\mathrm{avg}}$表示抗体群的平均亲和度值;$\lfloor\ \rfloor$表示向下取整。式(4.4.1)表明,抗体的克隆数与其自身的亲和度值成正比,进化的同时实现抗体的优胜劣汰。当抗体较差(亲和度值小于平均亲和度值)时,此抗体被抑制;当抗体较优(其亲和度值大于平均亲和度值)时,此抗体将被克隆。

自适应高频变异率是根据亲和度值及进化代数来调节抗体的高频变异概率的，其计算公式为[25]：

$$p_m = \begin{cases} p_{m_\max} - \left(\dfrac{p_{m_\max} - p_{m_\min}}{t_{\max}}\right)t, & \mathrm{Fit} > \mathrm{Fit}_{\mathrm{avg}} \\ p_{m_\max}, & \mathrm{Fit} \leqslant \mathrm{Fit}_{\mathrm{avg}} \end{cases} \tag{4.4.2}$$

式中，$p_m$ 表示需要变异抗体的高频变异概率；$p_{m_\max}$表示所给定的最大高频变异概率；$p_{m_\min}$表示所给定的最小高频变异概率；$t_{\max}$表示最大代数；$t$ 表示当前代数；Fit 表示需要克隆的抗体亲和度值；$\mathrm{Fit}_{\mathrm{avg}}$表示抗体群的平均亲和度值。式(4.4.2)表明，当抗体较差时，则其具有较高的变异概率；当抗体较优时，则当迭代次数越接近最大代数，抗体的变异概率就越小，根据其迭代状态赋予个体相应的变异概率。因此，在寻优开始时，变异概率较大，提高了全局搜索能力，使得抗体跳出局部极值点；随着迭代的进行，抗体的总变异概率逐渐减小，增强了局部搜索能力，使抗体更加接近全局最优解，有效防止进化早熟并增加抗体的多样性。

根据亲和度值及进化代数调节每代的更新数，自适应选择更新公式为[25]

$$d = \begin{cases} d_{\max} - \left(\dfrac{d_{\max} - d_{\min}}{t_{\max}}\right)t, & \mathrm{Fit}_{\max} - \mathrm{Fit}_{\min} < \dfrac{\mathrm{Fit}_{\mathrm{avg}}}{2} \\ d_{\max}, & \mathrm{Fit}_{\max} - \mathrm{Fit}_{\min} \geqslant \dfrac{\mathrm{Fit}_{\mathrm{avg}}}{2} \end{cases} \tag{4.4.3}$$

式中，$d$ 表示计算出的每代更新数；$d_{\max}$表示给定的最大更新数；$d_{\min}$表示给定的最小更新数；$\mathrm{Fit}_{\max}$和 $\mathrm{Fit}_{\min}$分别表示为抗体群中的最大和最小的抗体亲和度值。式(4.4.3)表明，抗体的平均亲和度值、最优抗体、最差抗体及进化过程都对每代更新率有很大影响。由于 $d$ 与抗体群的平均亲和度值成反比，因此随着抗体的不断进化，抗体亲和度值由寻优开始的较小(被更新的几率高)逐渐变大，更新率随之降低，有效防止了部分较优抗体被退化抗体替换的现象。

综上所述，基本的免疫克隆选择算法在求解问题之前需要设定参数，对于复杂问题的求解，参数的选择方法不同和参数产生的结论影响求解的结

果，而免疫克隆选择算法中的克隆概率 $C$、变异概率 $p_m$ 和选择概率 $d$ 都是固定的，不易取得最佳概率。因为全局搜索能力和局部搜索能力是相互矛盾的，用较小的变异概率来提高局部搜索能力，降低了全局搜索能力；而较大的变异概率，提高了全局寻优能力，但收敛精度下降。自适应方法主要是使算法在进化初期全局搜索能力较强，但随着进化的进行，全局搜索能力逐渐变弱，局部搜索能力逐渐增强，从而能够找到全局最优点，这样不仅增强了克隆概率、变异概率和选择概率在求解问题时的灵活性，也能有效防止“早熟”的产生。所以自适应免疫克隆算法对于均衡器最优权向量的搜索比标准免疫克隆算法更加有利。于是本节用克隆概率 $C$、变异概率 $p_m$ 和选择概率 $d$ 的自适应方法替换标准算法中反复试验的方法，有效优化了盲均衡算法的性能。

### 4.4.2 自适应免疫克隆选择算法优化正交小波超指数迭代常模盲均衡算法

将自适应免疫克隆算法引入到正交小波超指数迭代常模盲均衡算法中，得到的基于自适应免疫克隆优化的正交小波超指数迭代常模盲均衡算法(adaptive immune clonal selection optimization algorithm based orthogonal wavelet transform super-exponential Iterative constant modulus blind equalization algorithm，ACSA-WTSEI)[15]，如图 4.19 所示。自适应免疫克隆算法优化流程如图 4.20 所示。

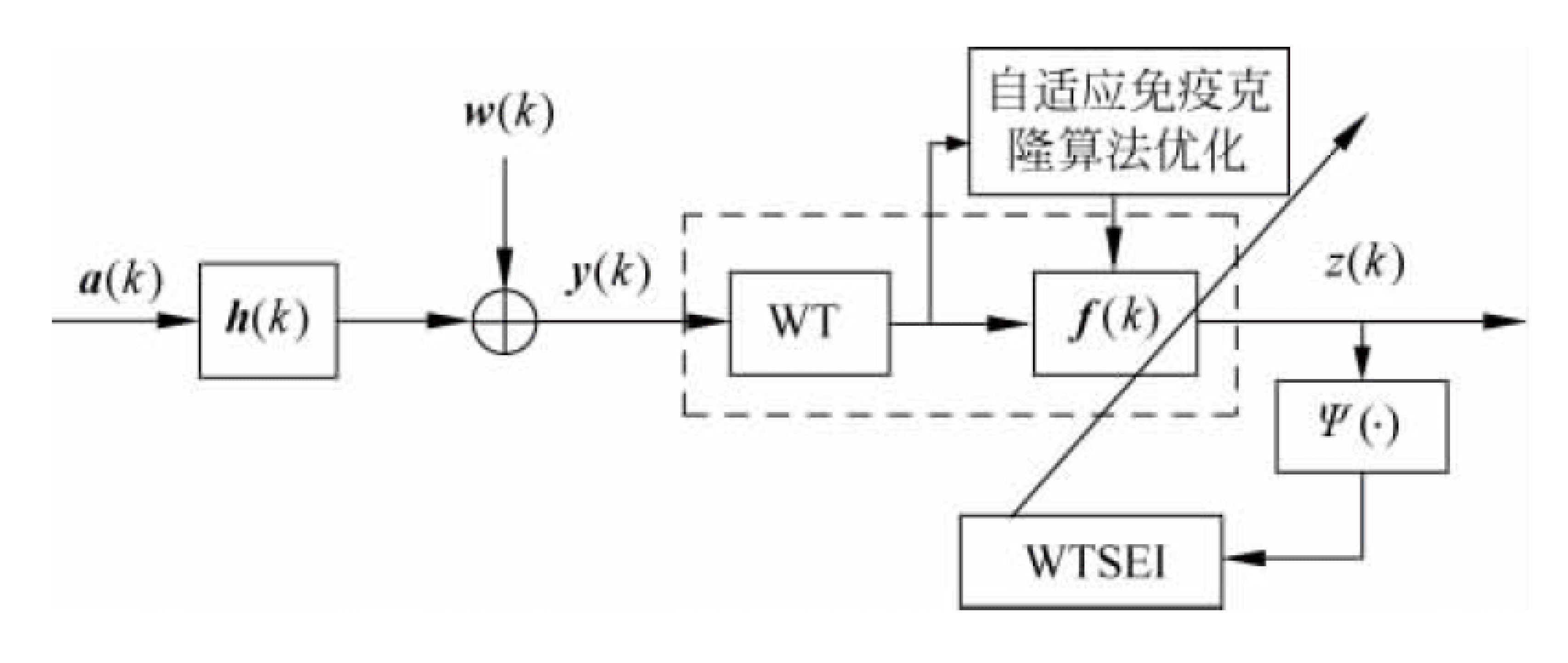



图 4.19 ACSA-WTSEI 原理

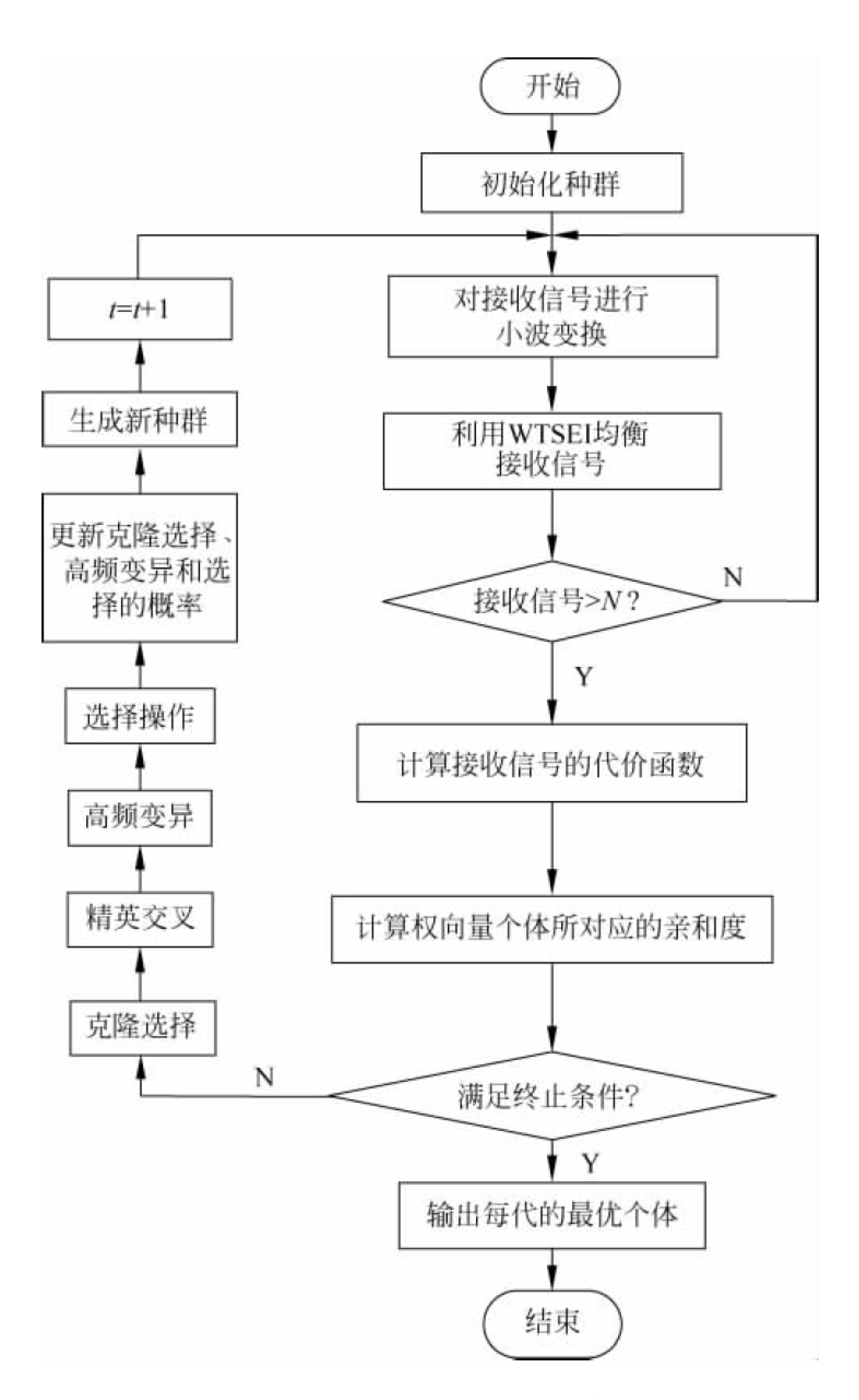



图 4.20 自适应免疫克隆优化算法流程

### 4.4.3 算法仿真

为了验证 ACSA-WTSEI 算法的有效性，将 WTSEI 算法和 CSA-WTSEI 算法作为比较对象，仿真参数设置如下。

(1) 标准克隆选择算法(SCSA)：抗体规模为 100，克隆控制因子为 0.6，变异概率为 0.1，算法最大迭代次数为 500。

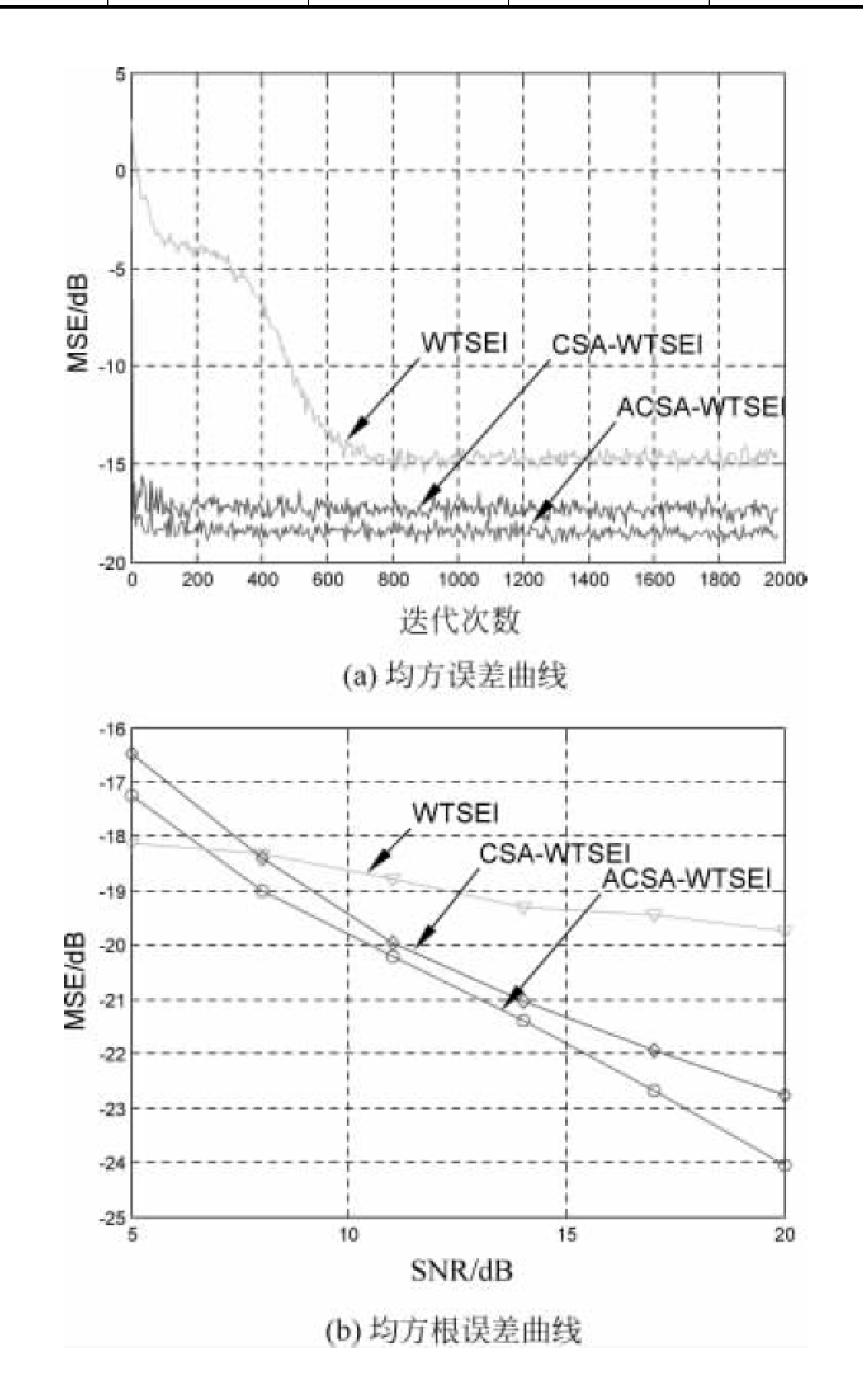
(2) 自适应克隆选择算法(ACSA)：抗体规模为 100，最大迭代次数为

500。$C_{max}=2$；$P_{m_max}=0.7$，$P_{m_min}=0.05$；$d_{max}=5$，$d_{min}=1$。

**【实验 4.4】** 当发射信号为 8PSK 时，信道 $\boldsymbol{h}=[-0.35\quad 0\quad 0\quad 1]$；信噪比为 20dB，均衡器长度均为 16。WTSEI 算法中，第 10 个抽头初始化设置为 1。其他参数设置，如表 4.1 所示。在保证眼图完全清晰睁开的条件下，500 次蒙特卡罗仿真结果，如图 4.21 所示。

表 4.1 仿真参数设置

| 算　　法 | 仿真步长 | 迭代步长 | 小波 | 分解层数 | $\beta$ 取值 |
|---|---|---|---|---|---|
| WTSEI | 0.015 | 0.02 | DB2 | 3 | 0.99 |
| CSA-WTSEI | 0.002 | 0.02 | DB2 | 3 | 0.9999 |
| ACSA-WTSEI | 0.002 | 0.02 | DB2 | 3 | 0.9999 |




(a) 均方误差曲线

(b) 均方根误差曲线

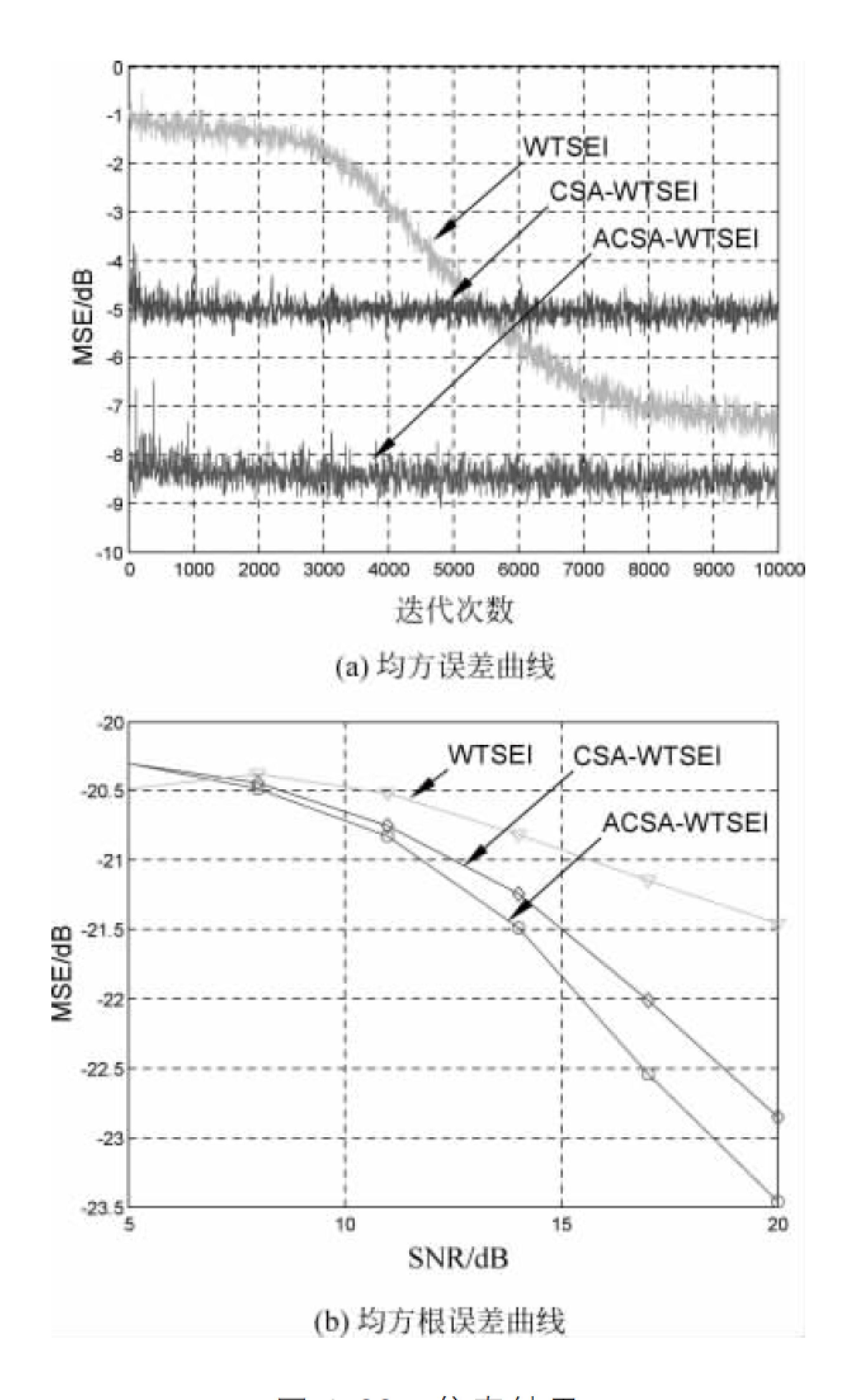
图 4.21 仿真结果

当发射信号为 16QAM 时，信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；信噪比为 20 dB，均衡器长度均为 16。WTSEI 算法中，第 3 个抽头初始化设置为 1。其他参数设置，如表 4.2 所示。

表 4.2　仿真参数设置

| 算　　法 | 仿真步长 | 迭代步长 | 小波 | 分解层数 | β 取值 |
|---|---|---|---|---|---|
| WTSEI | 0.0008 | 0.02 | DB2 | 3 | 0.99 |
| CSA-WTSEI | 0.000092 | 0.02 | DB2 | 3 | 0.9999 |
| ACSA-WTSEI | 0.000092 | 0.02 | DB2 | 3 | 0.9999 |

在保证眼图完全清晰睁开的条件下，500 次蒙特卡罗仿真结果，如图 4.22 所示。



(a) 均方误差曲线

(b) 均方根误差曲线

图 4.22　仿真结果

图 4.21(a)表明,在收敛速度上,ACSA-WTSEI 算法与 CSA-WTSEI 算法基本相同,但比 WTSEI 算法快了 700 步;在稳态误差上,ACSA-WTSEI 与 CSA-WTSEI 相比,减少了近 1dB;与 WTSEI 算法相比,减小了 4dB。图 4.21(b)表明,在均方误差与信噪比的关系曲线对比中,ACSA-WTSEI 算法性能也明显优于 CSA-WTSEI 算法和 WTSEI 算法。

图 4.22(a)表明,在收敛速度上,ACSA-WTSEI 算法与 CSA-WTSEI 算法基本相同,但比 WTSEI 算法快了 7500 步,在稳态误差上,ACSA-WTSEI 算法与 CSA-WTSEI 算法相比,减小了近 3.5dB,与 WTSEI 算法相比,减小了近 1dB。图 4.22(b)表明,在均方误差与信噪比的关系曲线对比中,ACSA-WTSEI 算法性能也明显优于 CSA-WTSEI 算法和 WTSEI 算法。

## 4.5 基于免疫克隆选择算法优化的正交小波支持向量机多模盲均衡算法

常数模盲均衡算法用于对不同模值的高阶 QAM 信号进行均衡时,其收敛速度慢、稳态误差大。多模盲均衡算法(MMA)能有效消除码间干扰并纠正星座图的相位旋转,但其收敛速度仍然较慢,稳态误差也较大[26-31]。文献[32-34]提出了一种利用支持向量机和结构风险最优化的特点,将支持向量机(SVM)引入盲均衡算法中,使算法具有全局最优解并且收敛速度也大大提高。但在 SVM 的构造过程中,其参数设置对最终分类精确度有较大的影响。合理的参数选择可以使 SVM 具有更高的精度、更好的泛化能力。本节利用免疫克隆选择算法的全局搜索能力,对 SVM 盲均衡算法的参数选择由人工选取变为自动确定,然后将 SVM 引入到正交小波多模盲均衡算法中,利用 SVM 对提取的一小段起始数据训练来估计盲均衡器初始权向量,利用免疫算法对 SVM 中的参数进行优化选择,并将 SVM 估计出的初始权向量作为正交小波多模盲均衡算法(WT-MMA)的权向量;针对标准免疫克隆算法对初始解有较强的依赖性,在迭代后期易出现停滞并

陷入局部最优的问题，本节提出了多种群免疫克隆选择算法，该算法提高了算法的整体搜索能力与搜索精度，有效弥补了标准免疫克隆选择算法搜索过程中的局部收敛问题，并由其优化正交小波多模盲均衡算法，得到了基于免疫优化支持向量机的正交小波多模盲均衡算法（immune optimization and SVM based WTMMA，IO-SVW-WTMMA）[35-36]。

1. 算法原理

基于免疫优化支持向量机的正交小波多模盲均衡算法原理如图 4.23 所示。

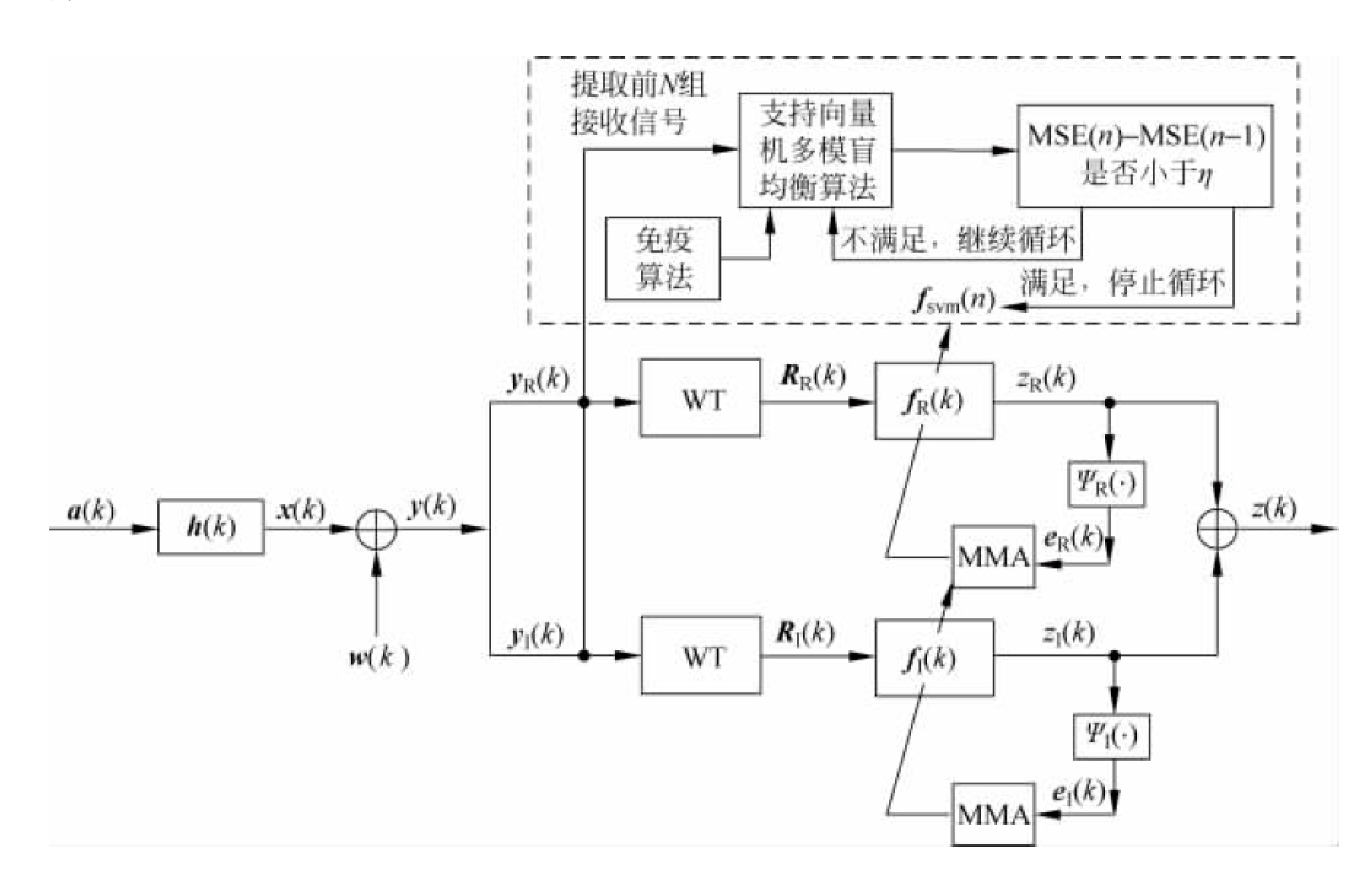



图 4.23 IO-SVW-WTMMA 原理

图 4.23 中，$\boldsymbol{a}(k)$为复信源发射信号，表示为 $\boldsymbol{a}(k)=\boldsymbol{a}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{a}_{\mathrm{I}}(k)$，$\boldsymbol{a}_{\mathrm{R}}(k)$和 $\boldsymbol{a}_{\mathrm{I}}(k)$分别为信源信号的实部和虚部；$\boldsymbol{h}(k)$为信道脉冲响应向量；$\boldsymbol{w}(k)$为加性高斯白噪声向量；$\boldsymbol{y}(k)=\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{y}_{\mathrm{I}}(k)$为均衡器的输入复信号向量；$\boldsymbol{\Psi}_{\mathrm{R}}(\cdot)$和 $\boldsymbol{\Psi}_{\mathrm{I}}(\cdot)$分别为误差函数 $e_{\mathrm{R}}(k)$，$e_{\mathrm{I}}(k)$的生成函数；$\boldsymbol{f}(k)$为均衡器权向量；$z(k)$为均衡器的输出序列。

$$\boldsymbol{y}(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+\boldsymbol{w}(k) \tag{4.5.1}$$

经过正交小波变换后均衡器输入信号由实部和虚部分别表示为

$$\boldsymbol{R}(k)=\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{R}_{\mathrm{I}}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{V}\boldsymbol{y}_{\mathrm{I}}(k)) \tag{4.5.2}$$

式中，$\boldsymbol{R}_{\mathrm{R}}(k)$和$\boldsymbol{R}_{\mathrm{I}}(k)$分别为$\boldsymbol{R}(k)$实部和虚部，且

$$\boldsymbol{R}_{\mathrm{R}}(k)=[u_{\mathrm{R}1,0}(k),u_{\mathrm{R}1,1}(k),\cdots,u_{\mathrm{R}J,k_J}(k),s_{\mathrm{R}J,0}(k),\cdots,s_{\mathrm{R}J,k_J}(k)]^{\mathrm{T}} \tag{4.5.3}$$

$$\begin{cases}u_{\mathrm{R}j,m}(k)=\sum\limits_{n=0}^{L-1}y_{\mathrm{R}}(k-n)\varphi_{j,m}(n)\\ s_{\mathrm{R}J,m}(k)=\sum\limits_{n=0}^{L-1}y_{\mathrm{R}}(k-n)\phi_{J,m}(n)\end{cases} \tag{4.5.4}$$

$$\boldsymbol{R}_{\mathrm{I}}(k)=[u_{\mathrm{I}1,0}(k),u_{\mathrm{I}1,1}(k),\cdots,u_{\mathrm{I}J,k_J}(k),s_{\mathrm{I}J,0}(k),\cdots,s_{\mathrm{I}J,k_J}(k)]^{\mathrm{T}} \tag{4.5.5}$$

$$\begin{cases}u_{\mathrm{I}j,m}(k)=\sum\limits_{n=0}^{L-1}y_{\mathrm{I}}(k-n)\varphi_{j,m}(n)\\ s_{\mathrm{I}J,m}(k)=\sum\limits_{n=0}^{L-1}y_{\mathrm{I}}(k-n)\phi_{J,m}(n)\end{cases} \tag{4.5.6}$$

式中，$k=0,1,\cdots,L-1$，$L=2^J$ 为均衡器的长度；$\varphi_{j,m}(n)$和$\phi_{J,m}(n)$分别表示小波函数和尺度函数；$u_{j,m}(k)$、$s_{J,m}(k)$分别为相应的小波和尺度变换系数；$\boldsymbol{V}$ 为正交小波变换矩阵。

均衡器输出为

$$\begin{cases}z_{\mathrm{R}}(k)=\boldsymbol{f}_{\mathrm{R}}^{\mathrm{H}}(k)\boldsymbol{R}_{\mathrm{R}}(k)\\ z_{\mathrm{I}}(k)=\boldsymbol{f}_{\mathrm{I}}^{\mathrm{H}}(k)\boldsymbol{R}_{\mathrm{I}}(k)\end{cases} \tag{4.5.7}$$

式中，$\boldsymbol{f}_{\mathrm{R}}^{\mathrm{H}}(k)$和$\boldsymbol{f}_{\mathrm{I}}^{\mathrm{H}}(k)$（H 表示共轭转置）分别为均衡器权向量的实部向量和虚部向量，$z_{\mathrm{R}}(k)$和$z_{\mathrm{I}}(k)$分别为均衡器输出信号的实部和虚部。

MMA 的代价函数为

$$J_{\mathrm{MMA}}=E\{(z_{\mathrm{R}}^2(k)-R_{\mathrm{R,MMA}}^2)^2+(z_{\mathrm{I}}^2(k)-R_{\mathrm{I,MMA}}^2)^2\} \tag{4.5.8}$$

式中，$R_{\mathrm{R,MMA}}^2=E\{(a_{\mathrm{R}}^4(k))\}/E\{(a_{\mathrm{R}}^2(k))\}$，$R_{\mathrm{I,MMA}}^2=E\{(a_{\mathrm{I}}^4(k))\}/E\{(a_{\mathrm{I}}^2(k))\}$。

均衡器的误差函数为

$$\begin{cases} e_{\mathrm{R,MMA}}(k) = z_{\mathrm{R}}(k)(z_{\mathrm{R}}^2(k) - R_{\mathrm{R,MMA}}^2) \\ e_{\mathrm{I,MMA}}(k) = z_{\mathrm{I}}(k)(z_{\mathrm{I}}^2(k) - R_{\mathrm{I,MMA}}^2) \end{cases} \tag{4.5.9}$$

均衡器权向量的迭代公式为

$$\begin{cases} \boldsymbol{f}_{\mathrm{R}}(k+1) = \boldsymbol{f}_{\mathrm{R}}(k) - \mu \hat{\boldsymbol{R}}^{-1}(k) e_{\mathrm{R,MMA}}(k) \boldsymbol{R}_{\mathrm{R}}^{*}(k) \\ \boldsymbol{f}_{\mathrm{I}}(k+1) = \boldsymbol{f}_{\mathrm{I}}(k) - \mu \hat{\boldsymbol{R}}^{-1}(k) e_{\mathrm{I,MMA}}(k) \boldsymbol{R}_{\mathrm{I}}^{*}(k) \end{cases} \tag{4.5.10}$$

式中，$\boldsymbol{R}^{*}(k)$为 $\boldsymbol{R}(k)$的共轭。

$$\hat{\boldsymbol{R}}^{-1}(k) = \mathrm{diag}[\sigma_{j,0}^2(k), \sigma_{j,1}^2(k), \cdots, \sigma_{J,k_j-1}^2(k), \sigma_{J+1,0}^2(k), \cdots, \sigma_{J+1,k_j-1}^2(k)] \tag{4.5.11}$$

式中，$\sigma_{j,n}^2(k)$、$\sigma_{J+1,k_j}^2(k)$分别表示对 $u_{j,m}(k)$、$s_{J,m}(k)$的平均功率估计，其递推估计公式为

$$\begin{cases} \hat{\sigma}_{j,m}^2(k+1) = \beta \hat{\sigma}_{j,m}^2(k) + (1-\beta) \mid u_{j,m}(k) \mid^2 \\ \hat{\sigma}_{J+1,m}^2(k+1) = \beta \hat{\sigma}_{J+1,m}^2(k) + (1-\beta) \mid s_{J,m}(k) \mid^2 \end{cases} \tag{4.5.12}$$

式中，$\beta$ 为平滑因子。式(4.5.2)～式(4.5.12)构成了基于正交小波变换的多模盲均衡算法(WTMMA)。

2. 支持向量机多模盲均衡算法

为了克服 WTMMA 利用代价函数对均衡器权向量的梯度求权向量缺乏全局搜索能力的缺点，本节利用支持向量机来搜寻最优化初始权向量，以弥补 WTMMA 的缺陷、避免搜索过程中陷于局部收敛的问题。

当发射器发射信号时，利用支持向量机对均衡器接收信号 $\boldsymbol{y}(k) = \boldsymbol{y}_{\mathrm{R}}(k) + \mathrm{j}\boldsymbol{y}_{\mathrm{I}}(k)(k=1,2,\cdots,N)$的前 $N$ 组向量进行均衡。根据发射信号的统计特性以及结构风险最小化原则，以精度 $\varepsilon$ 估计均衡器的初始权向量 $\boldsymbol{f}_{\mathrm{SVM}}(k)$。建立的支持向量机回归问题为

$$\min \quad E_{\mathrm{SVM}}(\boldsymbol{f}_{\mathrm{SVM}}(k)) = \frac{1}{2} \| \boldsymbol{f}_{\mathrm{SVM}}(k) \|^2 \tag{4.5.13}$$

约束函数为

$$\begin{cases} R^2(n) - [(\boldsymbol{f}_{\mathrm{SVM}}(k)]^{\mathrm{T}} \boldsymbol{y}(n))^2 \leqslant \varepsilon \\ [(\boldsymbol{f}_{\mathrm{SVM}}(k)]^{\mathrm{T}} \boldsymbol{y}(n))^2 - R^2(n) \leqslant \varepsilon \end{cases} \tag{4.5.14}$$

式中，$R^2(n)=R_R^2(n)+jR_I^2(n)$，参数 $\varepsilon$ 决定了 $\varepsilon$ 不敏感区域的宽度和支持向量的数目。

引入松弛变量 $\xi(n)$、$\tilde{\xi}(n)$ 和惩罚函数 $C$，式(4.5.13)和式(4.5.14)的最优化问题就可以转化为求解的约束最优化问题为

$$\min \quad E_{\mathrm{SVM}}(\boldsymbol{f}_{\mathrm{SVM}}(k)) = \frac{1}{2}\|\boldsymbol{f}_{\mathrm{SVM}}(k)\|^2 + C\sum_{n=1}^{N}(\xi(n)+\tilde{\xi}(n)) \tag{4.5.15}$$

约束条件为

$$\begin{cases} R^2(n)-([\boldsymbol{f}_{\mathrm{SVM}}(k)]^{\mathrm{T}}\boldsymbol{y}(n))^2 \leqslant \varepsilon+\xi(n) \\ ([\boldsymbol{f}_{\mathrm{SVM}}(k)]^{\mathrm{T}}\boldsymbol{y}(n))^2 - R^2(n) \leqslant \varepsilon+\tilde{\xi}(n) \\ \xi(n),\tilde{\xi}(n) \geqslant 0 \end{cases} \tag{4.5.16}$$

由于约束条件对于均衡器权向量 $\boldsymbol{f}_{\mathrm{SVM}}(n)$ 含有二次项，式(4.5.15)的最优化问题无法通过 SVM 所采用的线性规划方法求解。于是，根据文献[33]提出的一种迭代权二次规划算法(iterative reweighted quadratic programming，IRWQP)来解决这一问题，可以将式(4.5.16)中的二次约束改写为线性约束

$$\begin{cases} ((\boldsymbol{f}_{\mathrm{SVM}}(k))^{\mathrm{T}}\boldsymbol{y}(n))z_{\mathrm{SVM}}(n) - R^2(n) \leqslant \varepsilon+\tilde{\xi}(n) \\ R^2(n) - ((\boldsymbol{f}_{\mathrm{SVM}}(k))^{\mathrm{T}}\boldsymbol{y}(n))z_{\mathrm{SVM}}(n) \leqslant \varepsilon+\xi(n) \end{cases} \tag{4.5.17}$$

式中，$z_{\mathrm{SVM}}(k)=z_{\mathrm{RSVM}}(k)+jz_{\mathrm{ISVM}}(k)$。为导出原始问题的对偶问题，引入 Lagrange 函数

$$\begin{aligned} L(\boldsymbol{f}_{\mathrm{SVM}},\xi,\tilde{\xi},\alpha,\tilde{\alpha},b,\tilde{b}) = {} & \frac{1}{2}\|\boldsymbol{f}_{\mathrm{SVM}}(k)\|^2 + C\sum_{n=1}^{N}[\xi(n)+\tilde{\xi}(n)] - \\ & \sum_{n=1}^{N}(b_n\xi(n)+\tilde{b}_n\tilde{\xi}(n)) - \sum_{n=1}^{N}\alpha(n)[R^2(n) - \\ & ([\boldsymbol{f}_{\mathrm{SVM}}(k)]^2\boldsymbol{y}(n))z_{\mathrm{SVM}}(n)+\varepsilon+\xi(n)] - \\ & \sum_{n=1}^{N}\tilde{\alpha}(n)([\boldsymbol{f}_{\mathrm{SVM}}(k)]^2\boldsymbol{y}(n))z_{\mathrm{SVM}}(n) - R^2(n)+\varepsilon+\tilde{\xi}(n)] \end{aligned} \tag{4.5.18}$$

将式(4.5.13)～式(4.5.17)的原始最优化问题转换为凸二次规划问题(对偶问题)，即

$$\max \quad E'_{\mathrm{SVM}}=-\frac{1}{2}\sum_{n,j=1}^{N}(\tilde{\alpha}(j)-\alpha(j))(\tilde{\alpha}(n)-\alpha(n))(z_{\mathrm{SVM}}(j)z_{\mathrm{SVM}}(n))K\langle \boldsymbol{y}_j,\boldsymbol{y}_n\rangle-\varepsilon\sum_{j=1}^{N}(\tilde{\alpha}(j)+\alpha(j))+\sum_{j=1}^{N}(\tilde{\alpha}(j)-\alpha(j)) \tag{4.5.19}$$

约束条件为

$$\begin{cases}\sum_{j=1}^{N}(\tilde{\alpha}(j)-\alpha(j))=0\\ 0\leqslant\tilde{\alpha}(j)\leqslant C,j=1,2,\cdots,N\end{cases} \tag{4.5.20}$$

通过比较原始问题与对偶问题，均衡器权向量计算公式为

$$\boldsymbol{f}_{\mathrm{SVM}}(k)=\sum_{n=1}^{N}(\tilde{\alpha}(n)-\alpha(n))z_{\mathrm{SVM}}(n)\boldsymbol{y}(n) \tag{4.5.21}$$

式中，Lagrange 乘子 $\tilde{\alpha}(n)$和 $\alpha(n)$通过式(4.5.19)和式(4.5.20)来求解。

通过上述过程，便可以计算出均衡器初始权向量 $\boldsymbol{f}_{\mathrm{SVM}}(k)$，再进行循环迭代直至满足切换条件。$\boldsymbol{f}_{\mathrm{SVM}}(k)$的更新公式为

$$\boldsymbol{f}_{\mathrm{SVM}}(k)=\lambda\boldsymbol{f}_{\mathrm{SVM}}(k-1)+(1-\lambda)\boldsymbol{f}_{\mathrm{SVM}}(k-2) \tag{4.5.22}$$

式中，$k$ 为迭代次数，$\lambda$ 为迭代步长。

当满足切换条件

$$\begin{cases}\mathrm{MSE}(k)=\frac{1}{N}\sum_{n=1}^{N}(|z_{\mathrm{SVM}}(n)|^2-R(n))\\ |\mathrm{MSE}(k)-\mathrm{MSE}(k-1)|\leqslant\eta\end{cases} \tag{4.5.23}$$

时，可得全局最优的初始化权向量 $\boldsymbol{f}_{\mathrm{SVM}}(k)$，将其作为 WTMMA 的初始化权向量。式(4.5.23)中，$\eta$ 为切换阈值。

### 3. 算法仿真

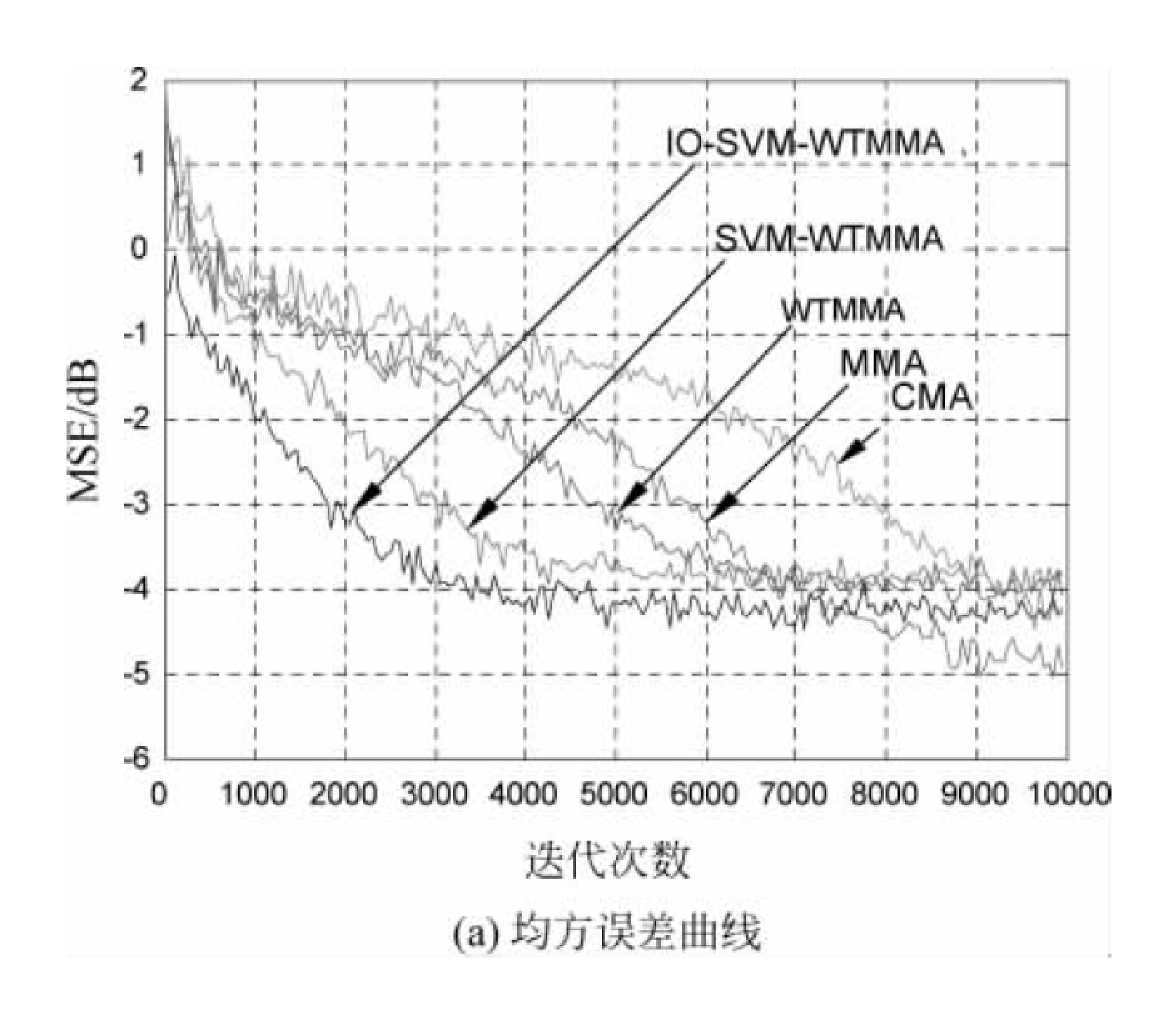
为了验证 IO-SVM-WTMMA 的有效性，以 CMA、MMA、WTMMA 和

SVM-WTMMA 作为比较对象,进行仿真实验。仿真实验中,抗体规模为 100,克隆控制因子为 0.6,精英交叉概率为 0.2,变异概率为 0.1,最大迭代次数为 200。参数 $C$ 和 $\varepsilon$ 优化取值范围为 $1\leqslant C\leqslant 20$,$0.000\,01\leqslant\varepsilon\leqslant 0.1$,支持向量机初始化所提取的训练样本个数 $N=2000$。

**【实验 4.5】** 发射信号为 128QAM,信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$;均衡器权长为 16,信噪比 30dB。SVM-WTMMA 中,$C=15$,$\varepsilon=0.1$;IO-SVM-WTMMA 中,免疫优化选择最优参数为 $C=17.8477$,$\varepsilon=0.0765$。其他参数设置,如表 4.3 所示。1000 次蒙特卡罗仿真结果如图 4.24 所示。

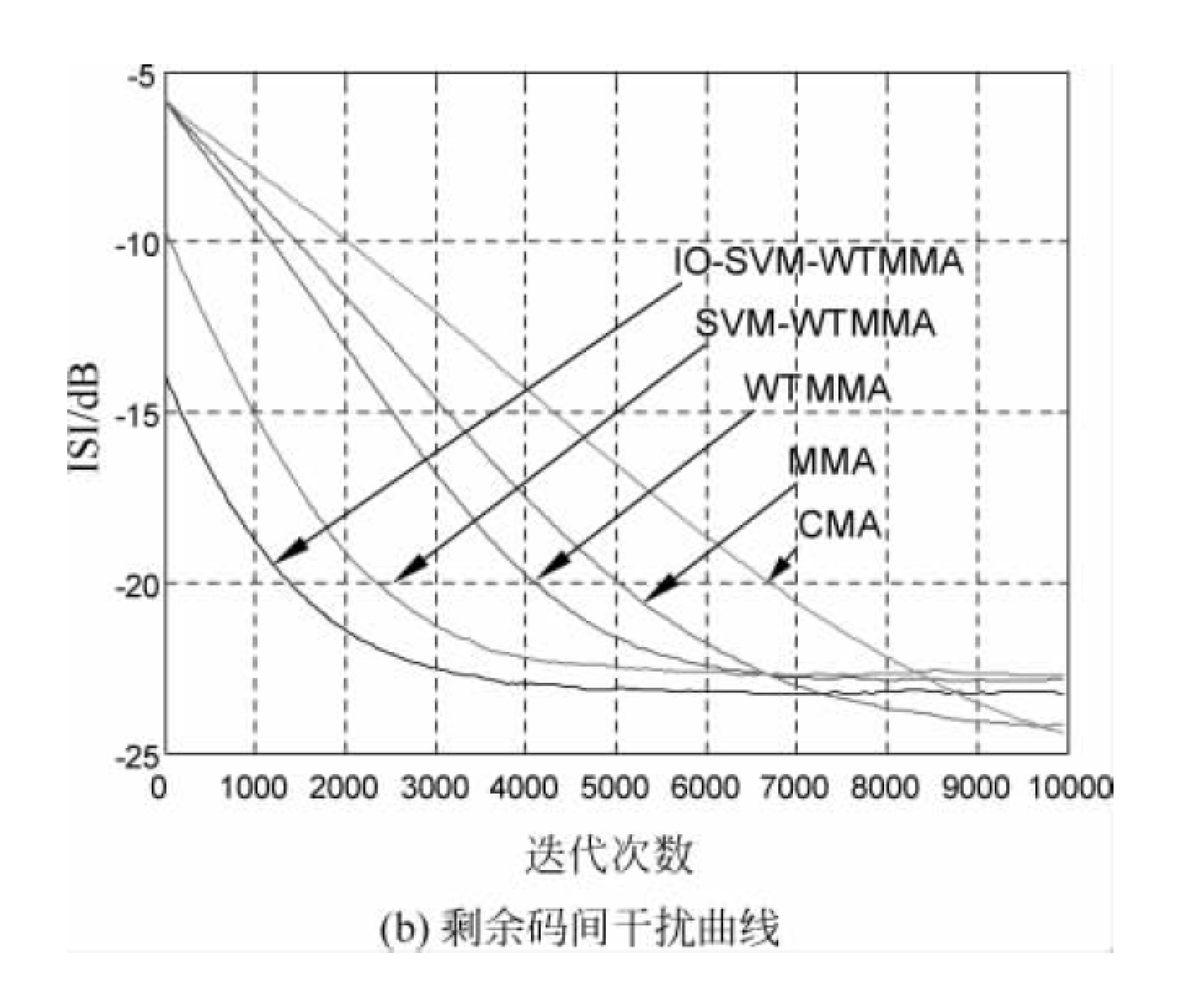
表 4.3 仿真参数

| 算法 | 仿真步长 | 小波 | 分解层数 | 功率初始值 | $\beta$ 取值 | 初始化权值 |
|---|---|---|---|---|---|---|
| CMA | 0.000 000 1 | | | | | 第 8 个抽头系数为 1,其余为 0 |
| MMA | 0.000 000 35 | | | | | |
| WT-MMA | 0.000 018 | DB1 | 2 | 8 | 0.95 | |
| SVM-WTMMA | 0.000 013 | DB1 | 2 | 8 | 0.99 | |
| IO-SVM-WTMMA | 0.000 012 | DB1 | 2 | 20 | 0.9999 | |

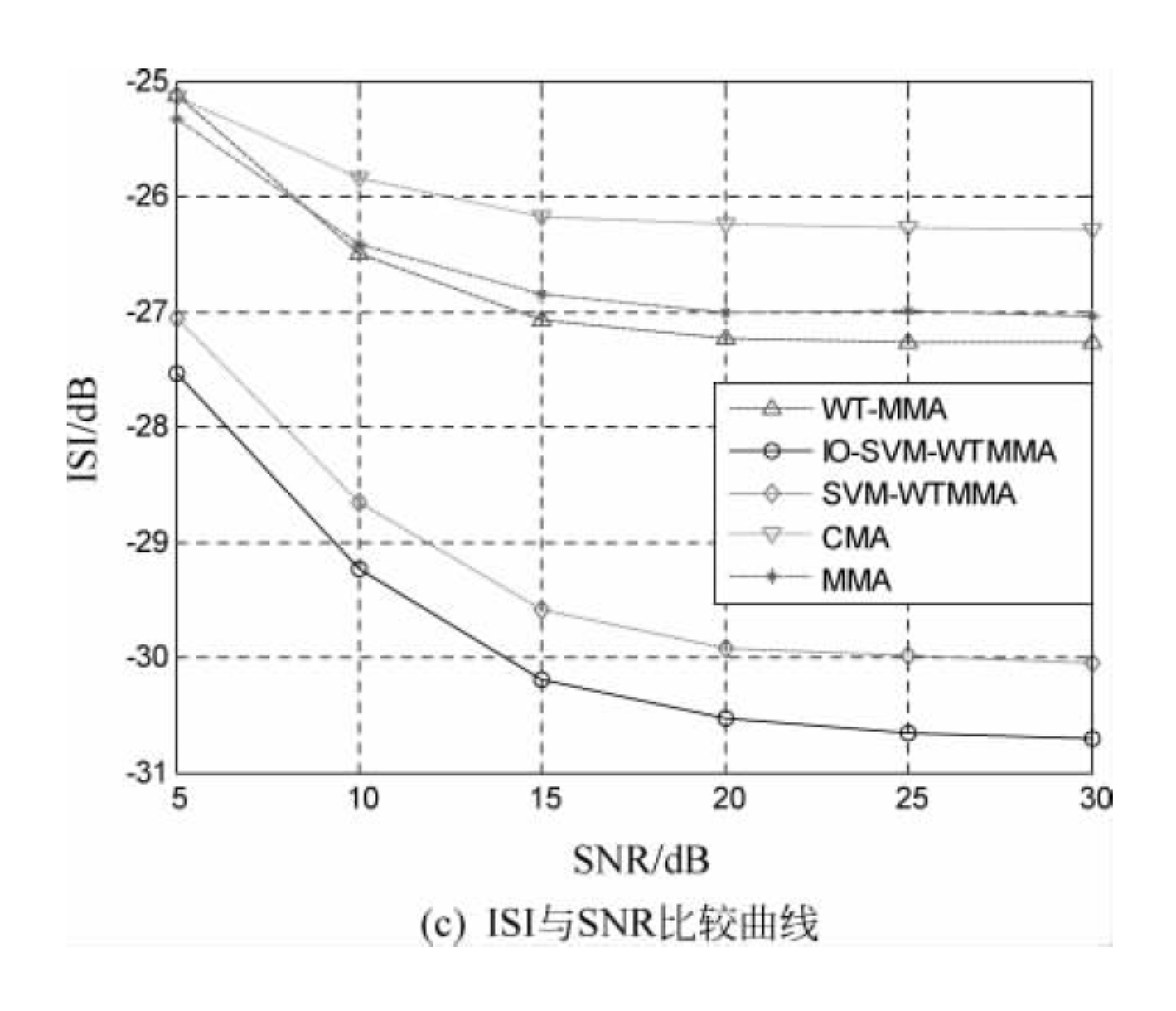


(a) 均方误差曲线

图 4.24 仿真结果

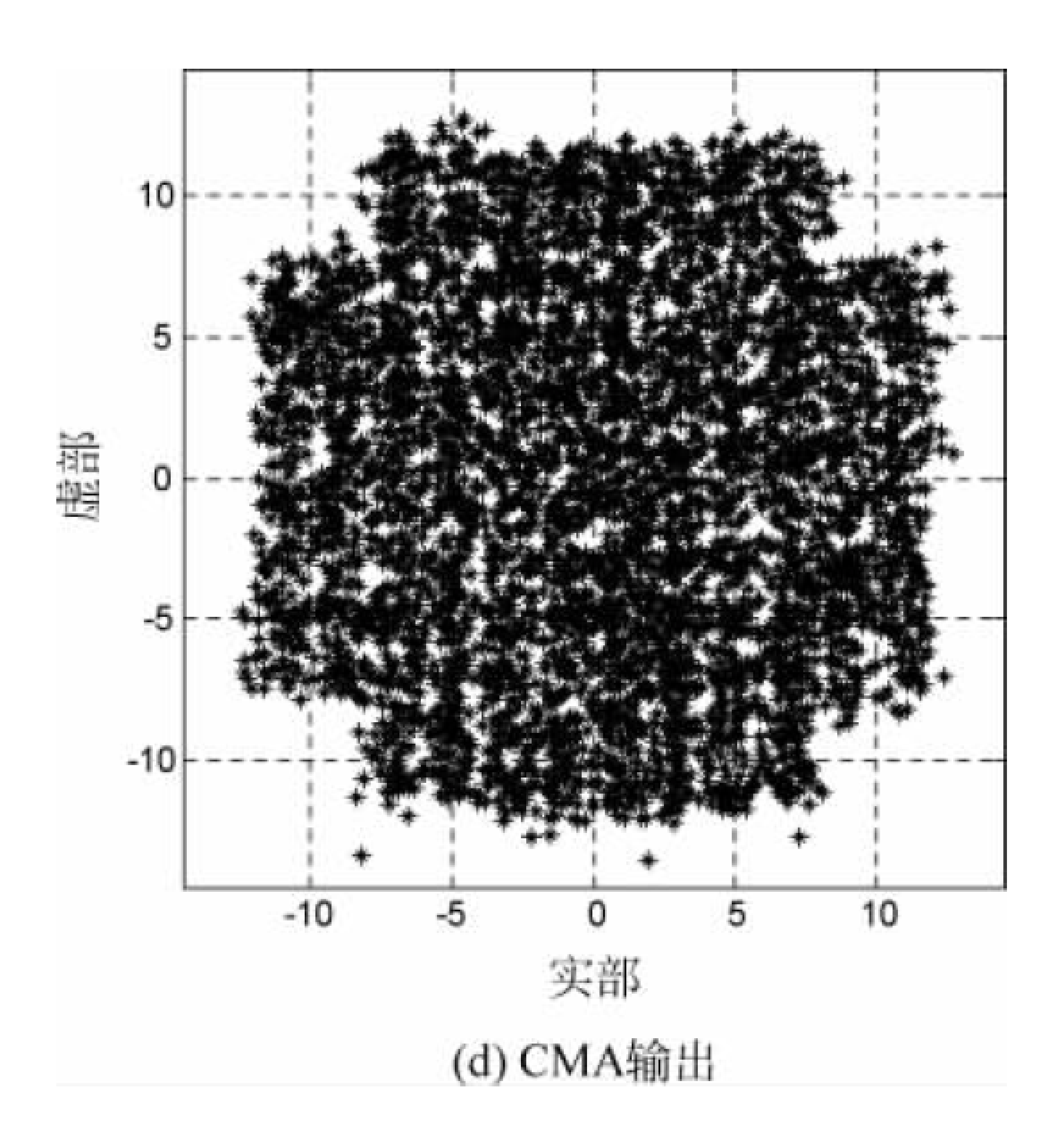


(b) 剩余码间干扰曲线

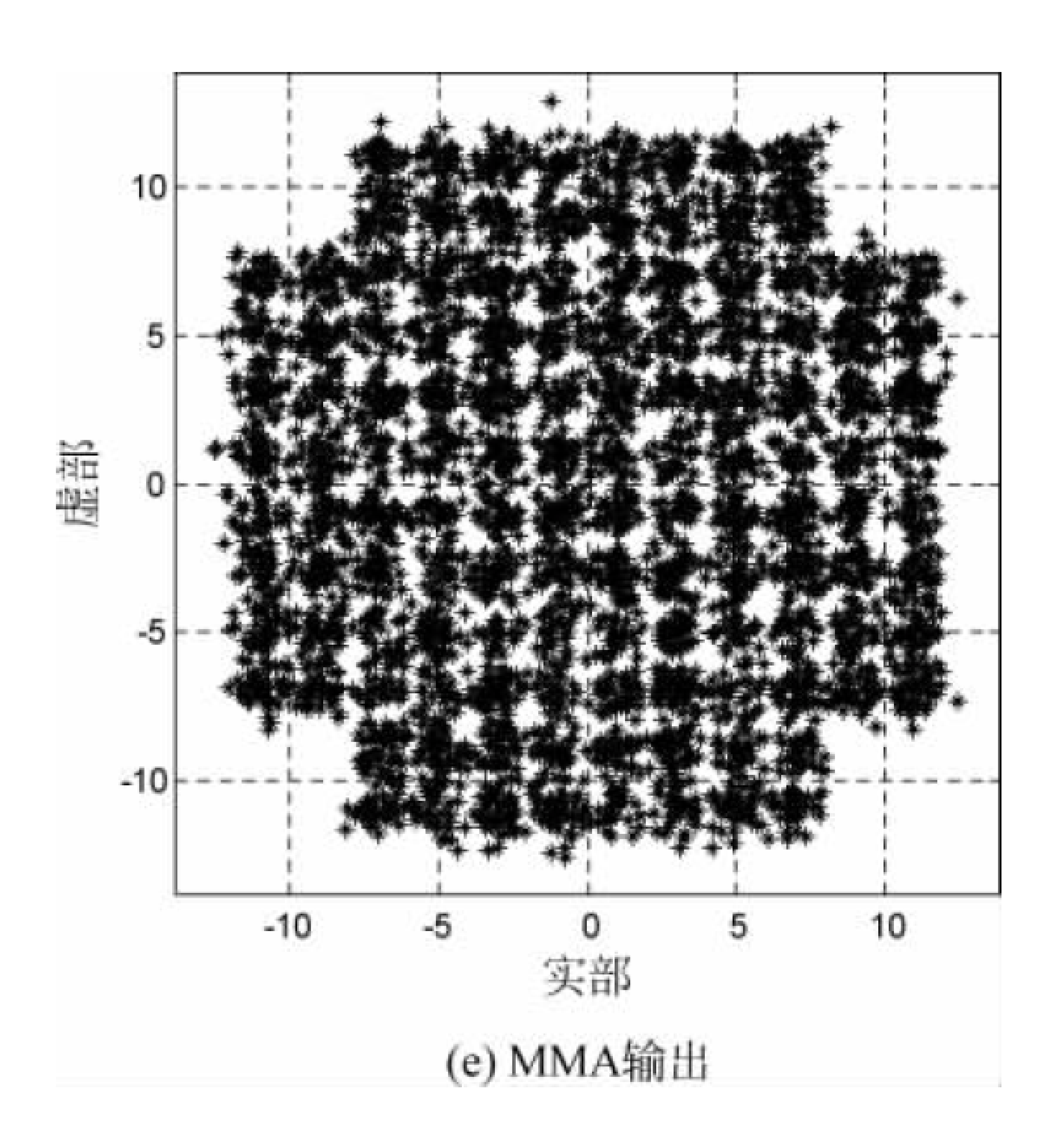


(c) ISI与SNR比较曲线

图 4.24 （续）

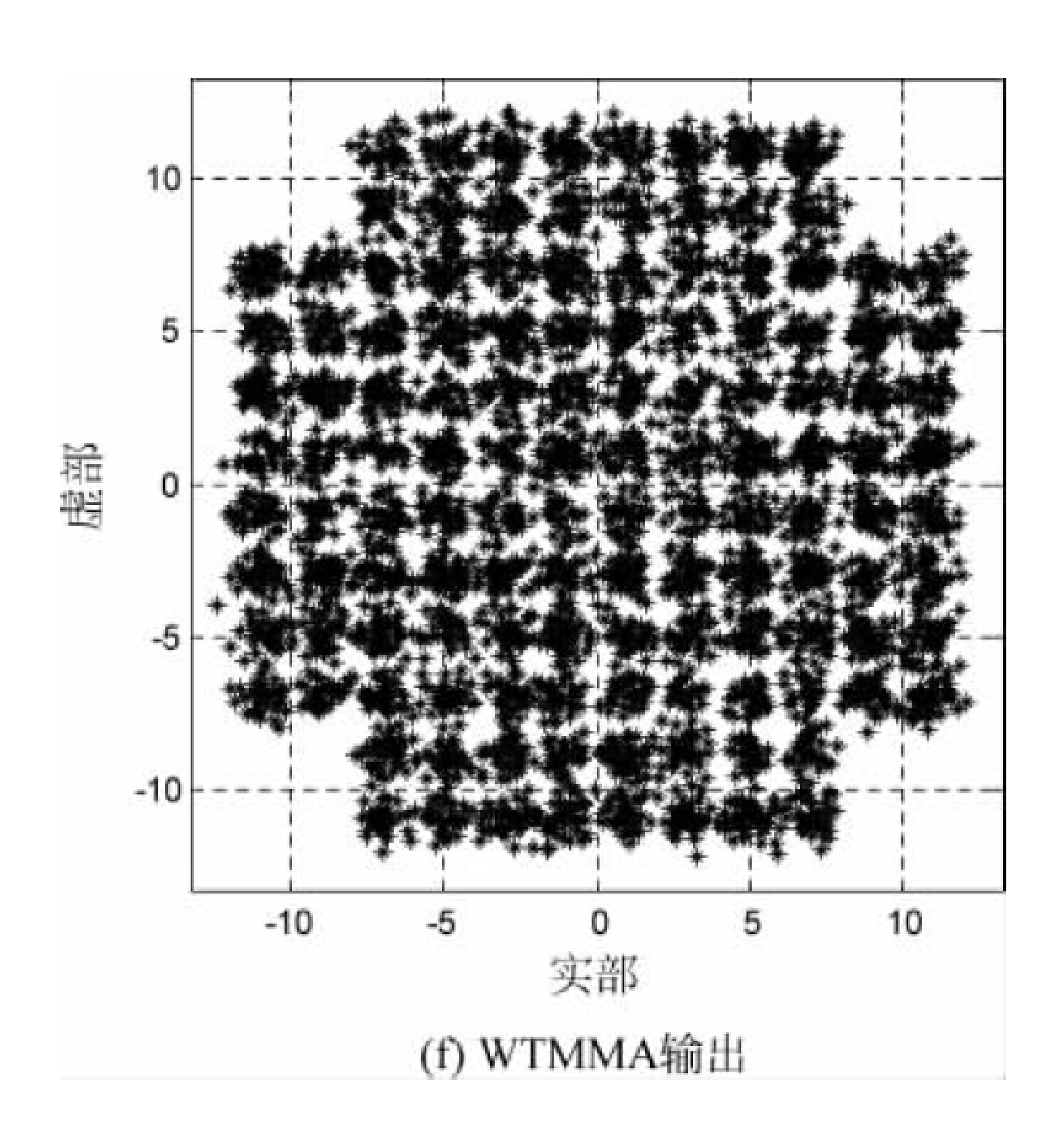


(d) CMA输出

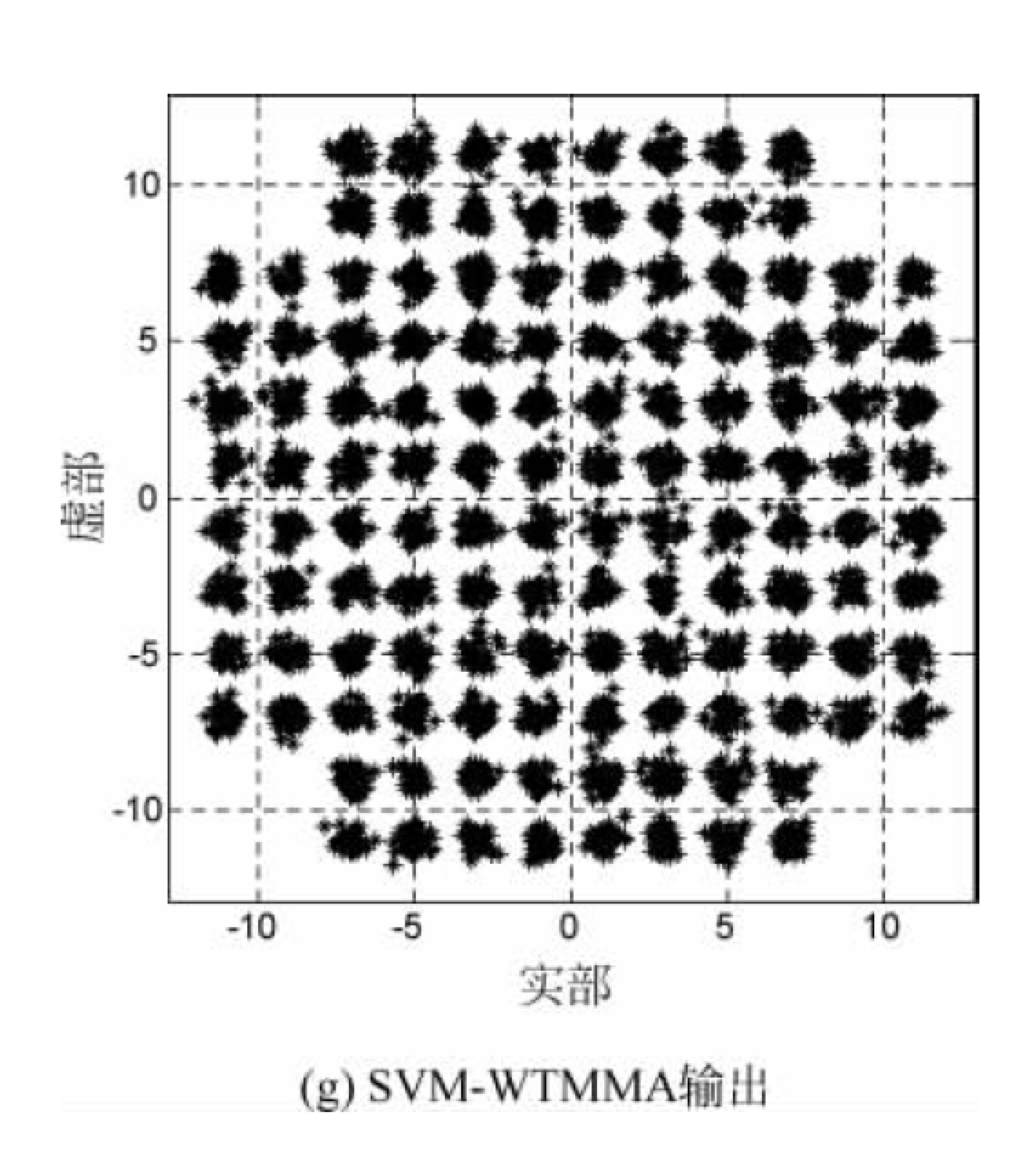


(e) MMA输出

图 4.24 （续）

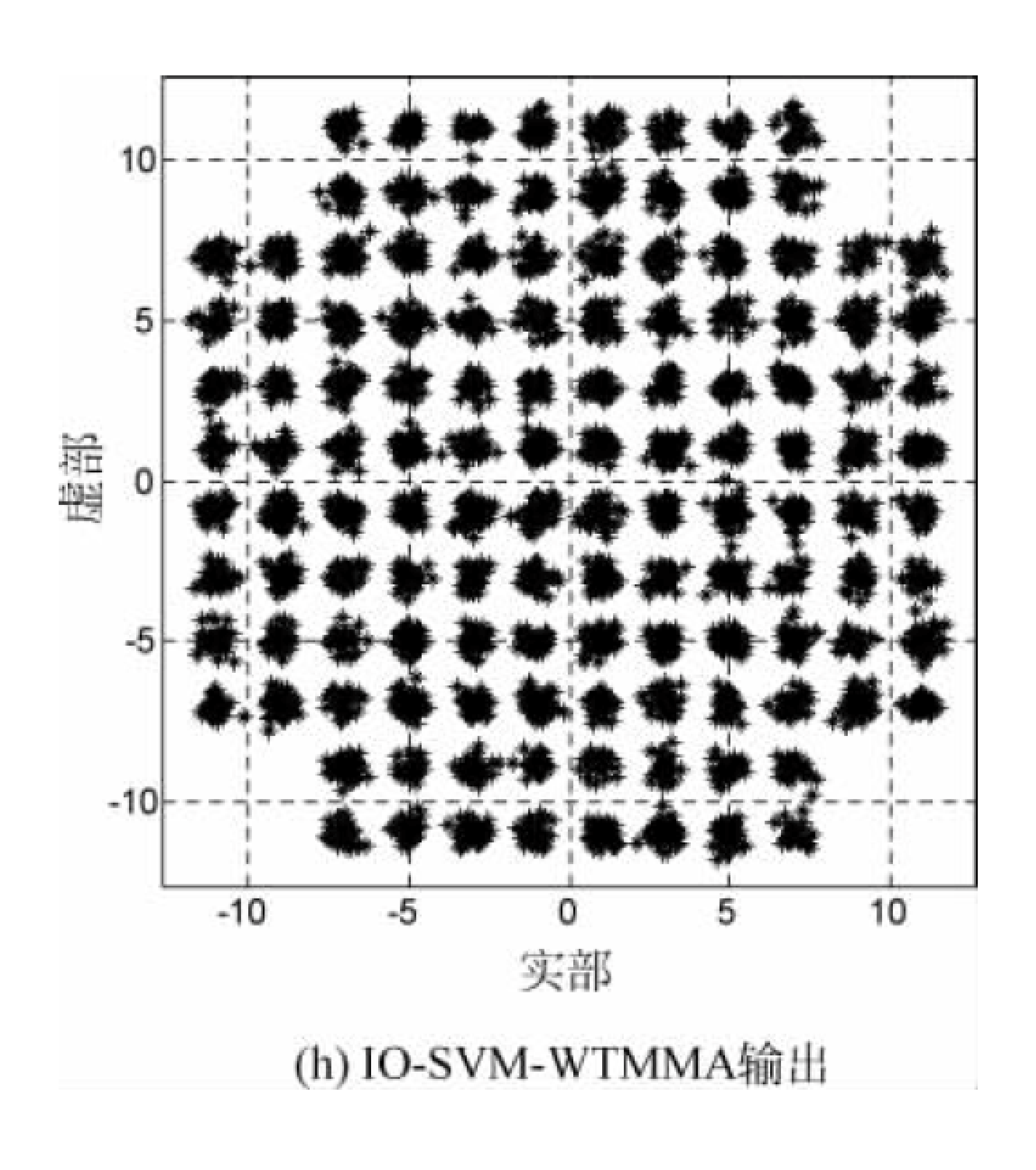


(f) WTMMA输出



(g) SVM-WTMMA输出

图 4.24 （续）



(h) IO-SVM-WTMMA输出

图 4.24 （续）

图 4.24(a)、(b)表明：在收敛速度上，IO-SVM-WTMMA 与 SVM-WTMMA 基本相同，但比 MMA 快了近 6000 步、比 WTMMA 快了近 3000 步；在剩余码间干扰上，IO-SVM-WTMMA 比 WTMMA 和 SVM-WTMMA 减小近 0.8dB。图 4.24(c)表明，随着信噪比的增加，5 种算法的剩余码间干扰都在不断减小，而 IO-SVM-WTMMA 减小的幅度最大，同一信噪比比较更能体现该算法的优越性。图 4.24(e)～(h)表明，IO-SVM-WTMMA 的星座图比 CMA、MMA、WTMMA 和 SVM-WTMMA 更加清晰、紧凑，有很强的抗码间干扰能力，具有一定的实用性。

## 4.6 基于多种群免疫克隆选择算法优化的正交小波多模盲均衡算法

### 4.6.1 多种群免疫克隆选择算法

传统的免疫算法对初始解有较强的依赖性，容易陷入局部最优的平衡态，在迭代后期容易出现停滞现象，导致搜索精度不高，这使得算法最终搜

索得到的结果往往不是全局最优解，而是局部最优解[37,38]。为了使免疫算法避免过早收敛，本节提出的多种群免疫克隆算法是受多种群遗传算法思想[39,40]的启发，在免疫克隆算法并行运算基础上，通过引入移民算子进行子种群间的交换信息（一般为最优个体）以及多种群并行进化的思想来增加基因模式数，将免疫克隆算法分解为在多个子种群间分别进行独立的免疫操作，并采用传优算子把当前最优抗体分配到各个子群体中，避免未成熟收敛。多种群免疫克隆算法的基本思想是：利用多种群并行进化的框架，在相同搜索空间上，以一定方式生成多个抗体种群代替原来的单一种群，每个抗体种群通过赋予不同的控制参数来实现不同的搜索目的。算法中用多个子种群代替原始种群在可行解域进行搜索，将每个子种群按各自不同的进化策略和免疫操作并行进化。这种方法可以有效提高多种群免疫克隆算法的优化效率，加快进化速度，保持最优个体进化稳定性。

在自然界中，无论是何种生物，不同地域有不同的特点和不同的进化程度，其适应能力上的强弱也各有不同，虽然优良个体只占整体种群的一小部分，但比普通个体拥有更多的资源，同时优良个体繁殖所产生的新个体通常具有更加优越的能力。本节提出的多种群免疫克隆算法就是借鉴自然界生物中的这一现象设计出来的。整个算法由各种群分别进行独立的进化进程，从环境中得到计算资源，构成一个由多个普通种群和一个精英种群的计算环境。各个普通种群将进化得到的优良个体进入精英种群，以分别改善各个种群的多样性[41,42]。系统结构，如图 4.25 所示。

由前面的分析可知，提出的多种群免疫克隆算法（multiple population CSA，MPCSA）可用来弥补基本免疫克隆选择算法所存在的缺陷。

MPCSA 在 CSA 的基础上主要提出了以下几个新概念：

(1) 突破 CSA 仅靠单个群体进行免疫进化的框架，引入多个种群同时进行优化搜索；不同的种群赋以不同的控制参数，实现不同的搜索目的。

(2) 各个种群之间通过移民算子进行信息交换，实现多种群协同进化；最优解的获取是多个种群协同进化的综合结果。

(3) 通过人工选择算子保存各种群中每次进化代的最优个体，作为判

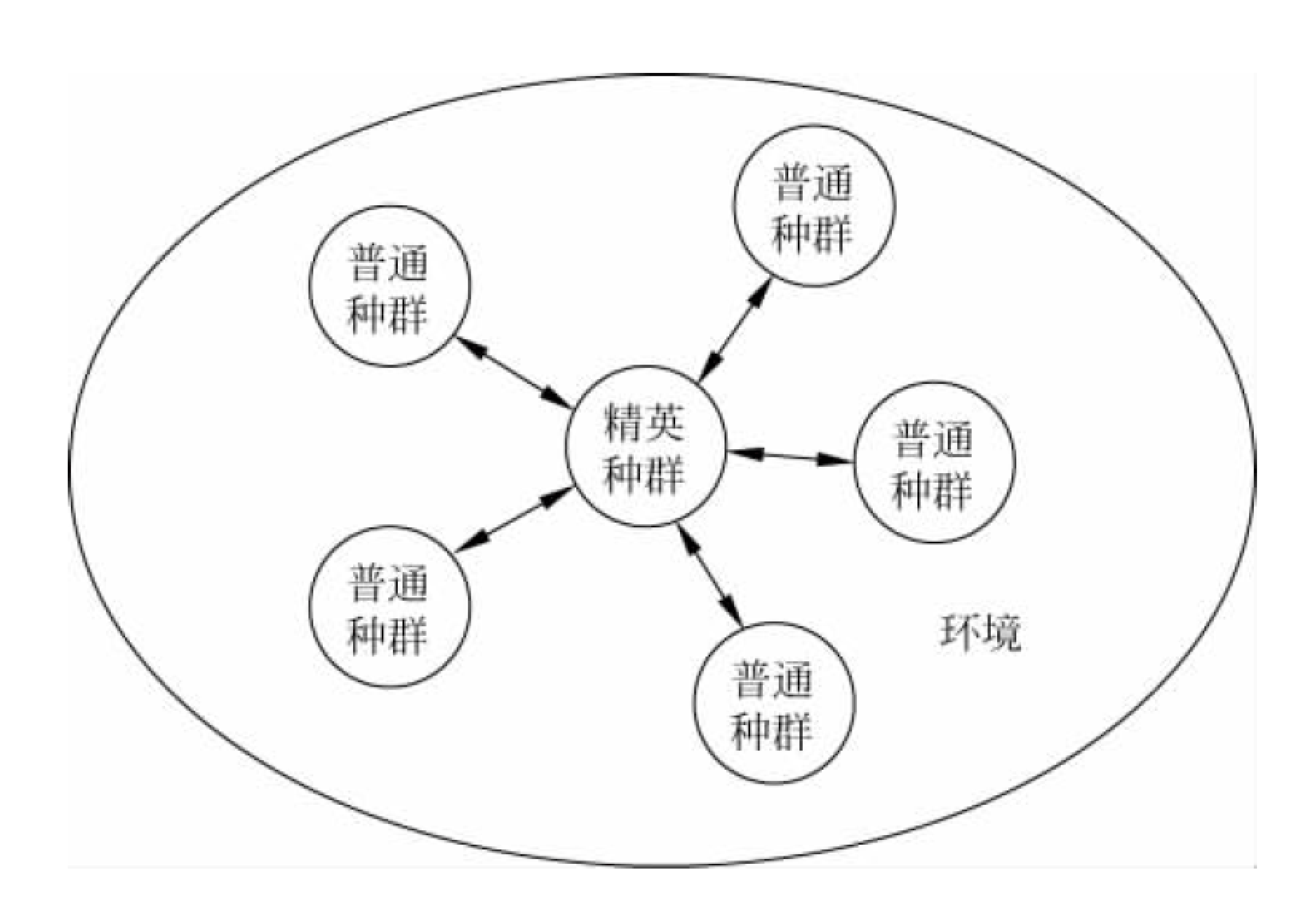



图 4.25　多种群系统

断算法收敛的依据。

MPCSA 算法结构如图 4.26 所示。

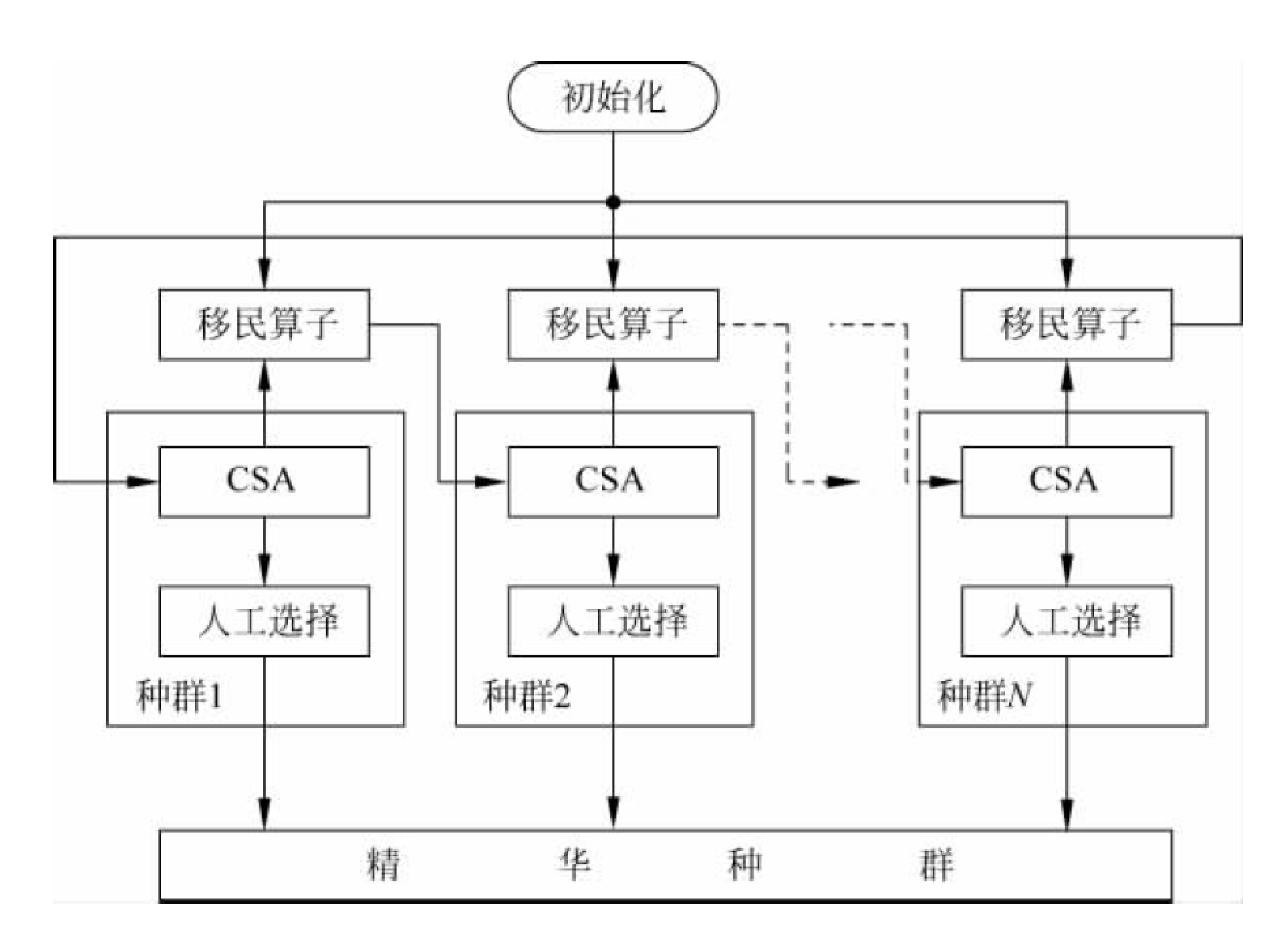



图 4.26　MPCSA 算法结构

图中，种群 1～$N$ 的进化机制都是采用常规的 CSA 克隆选择、精英交叉和高频变异。

由于各抗体种群的搜索目的是不同的，因此算法全局搜索和局部搜索

能力的均衡主要决定于控制参数交叉概率 $p_c$ 和变异概率 $p_m$ 的取值。在 SCSA 中，$p_c$，$p_m$ 取值较大时，更加侧重于全局搜索，算法产生新个体的能力也随之强，反而导致局部细化搜索能力不足；$p_c$，$p_m$ 取值较小时，全局搜索能力同时减弱，算法使个体趋于收敛的能力越强，局部搜索能力却得以增强。因此，建议选取较大的 $p_c$（0.7～0.9）和较小的 $p_m$（0.01～0.1）。MPCSA 有效地弥补了单克隆选择算法（single clonal selection algorithm，SCSA）的不足，通过多个设有不同控制参数的种群协同进化，以搜索不同解空间中的最优解，同时使全局搜索和局部搜索得到了兼顾。

各抗体种群是相对独立的，种群之间相互交换信息是通过移民算子将各种群在进化过程中出现的最优个体定期地（每隔一定的进化代数）引入其他种群中。其操作规则是：将目标种群中的最差个体用源种群的最优个体代替（图 4.26 中，移民算子输入端种群称为源种群，输出端种群称为目标种群）。如果在 MPGA 中没有移民算子，就失去了 MPCSA 的特色，等于用不同的控制参数进行多次 SCSA 计算，各种群之间失去了信息交换。

多种群免疫克隆选择算法中的精华种群不同于其他种群，为了保证进化过程中各种群产生的最优个体不被破坏和丢失，精华种群不进行克隆、交叉、变异等免疫操作。在进化过程中的每一代，都通过人工选择算子选出每个种群的最优个体放入精华种群中保存，其作用在于尽可能保护最优个体加快收敛速度。同时，精华种群也是判断算法终止的重要依据。

### 4.6.2 基于多种群免疫克隆选择优化的正交小波多模盲均衡算法

为了进一步提高算法的性能，增强算法的实用性，避免不当的均衡器权向量初始化所造成算法收敛至局部极小值，甚至发散，因此在正交小波多模盲均衡算法基础上，提出了一种多种群免疫克隆优化的正交小波多模盲均衡算法[11]，该算法的基本思想是：由于正交小波多模盲均衡算法是基于梯度信息对均衡器权向量进行更新的，全局搜索能力较差，易陷入局部收敛。

本节利用多种群免疫克隆算法的全局搜索性能来改变 WTMMA 依赖梯度信息对均衡器权向量进行调整，以克服 WTMMA 的缺陷。这样得到的基于多种群免疫克隆优化的正交小波多模盲均衡算法（multiple population CSA based WTCMA，MPCSA-WTMMA）。其基本结构如图 4.27 所示。

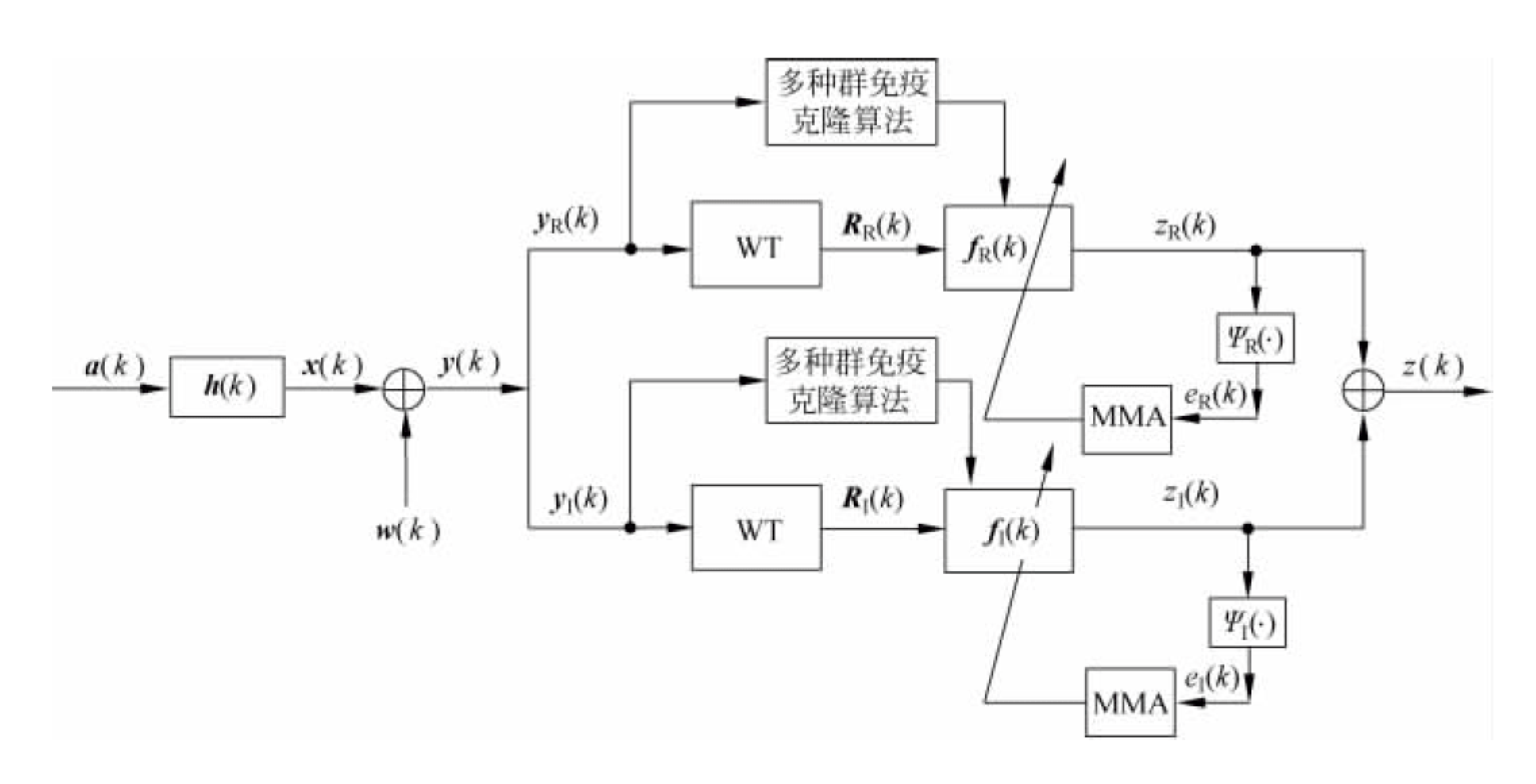



图 4.27　MPCSA-WTMMA 原理

MPCSA 优化均衡器权向量的实现步骤如下。

步骤 1：种群初始化。首先，随机生成 $N$ 个子种群，各子种群规模数同为 $M$，所以初始种群中共有 $N\times M$ 个体。并且，设置进化代数计数器 $t=0$ 和最大进化代数 $T$；各子种群的交叉概率为 $p_c$，变异概率为 $p_m$。

步骤 2：各子种群独立进行免疫克隆选择操作：(a)克隆选择操作；(b)精英交叉操作；(c)高频变异操作。

步骤 3：移民算子。通过人工选择算子设定的频度将各种群的优良个体保存到精华种群；移民算子原理就是在每一代进化过程中以一定的淘汰率将最差的个体淘汰，然后用产生的新个体代替。由于各抗体种群是相对独立的，种群之间相互交换信息是通过移民算子将各种群在进化过程中出现的最优个体定期地（每隔一定的进化代数）引入至其他种群中。本节用当前的最优个体去代替每个种群中的最差个体，这样不仅充分保留了父代个体好的基因模式，又保证了群体的多样性，提高了免疫克隆算法的寻优性能。

多种群免疫克隆算法优化流程如图 4.28 所示。

开始
随机初始化多个种群
对接收信号进行小波变换
利用MMA均衡接收信号
接收信号>N?
N
Y
计算接收信号的代价函数值
计算每个权向量个体所对应的亲和度
满足终止条件？
N
Y
输出每代最优个体
结束
移民算子
克隆选择
精英交叉
高频变异
人工选择操作
生成精华种群
t=t+1

图 4.28　多种群免疫克隆算法优化流程

### 4.6.3　算法仿真

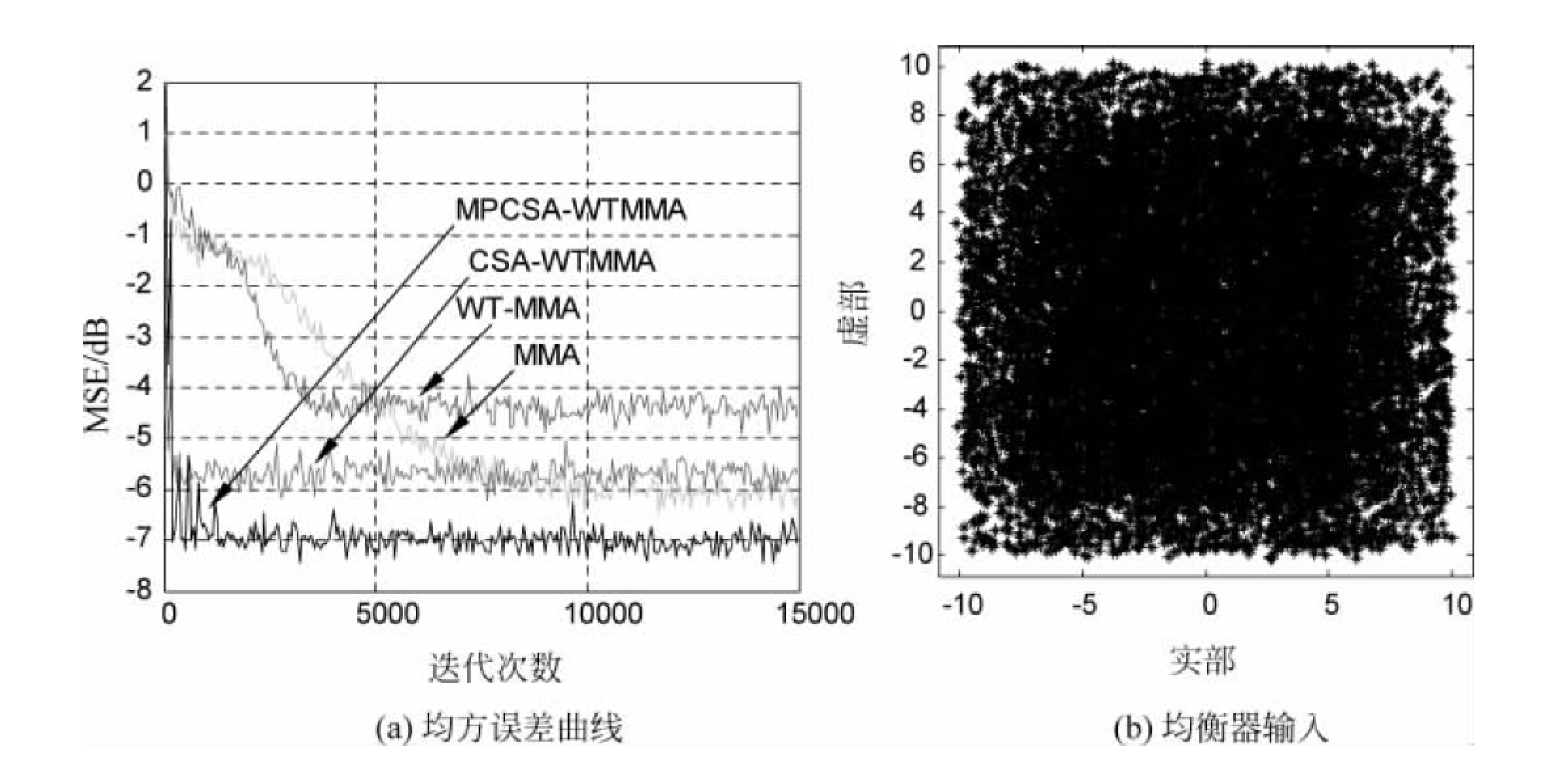
为了检验 MPCSA-WTMMA 的有效性，以 MMA、WTMMA 和 CSA-WTMMA 作为比较对象，进行仿真实验。仿真试验中，抗体规模为 100，克隆控制因子为 0.6，最大迭代次数为 100。

**【实验 4.6】** 发射信号为 64QAM，信道 $\boldsymbol{h}=[-0.35\quad 0\quad 0\quad 1]$；均衡器权长均为 16，信噪比为 25dB。其他参数设置如表 4.4 所示。500 次蒙特

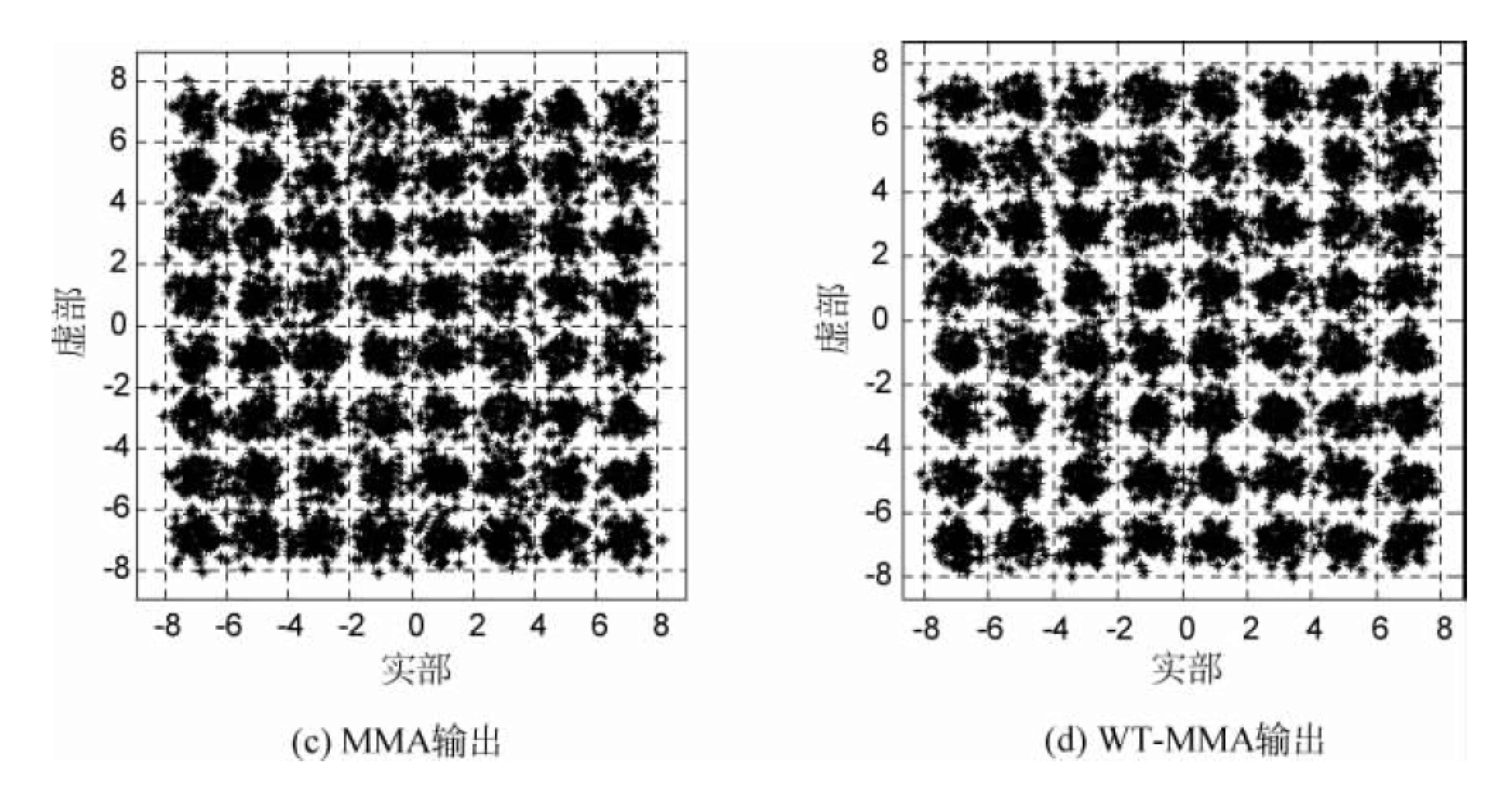
卡罗仿真结果如图 4.29 所示。

表 4.4 仿真参数

| 算　　法 | 仿真步长 | 小波 | 分解层数 | 功率初始值 | β 取值 | 初始化权值 |
|---|---|---|---|---|---|---|
| MMA | 0.000 001 | | | | | 第 12 个抽头系数为 1,其余为 0 |
| WT-MMA | 0.000 07 | DB2 | 3 | 4 | 0.999 | 第 6 个抽头系数为 1,其余为 0 |
| CSA-WTMMA | 0.000 012 | DB2 | 3 | 30 | 0.9999 | |
| MPCSA-WTMMA | 0.000 01 | DB2 | 3 | 30 | 0.9999 | |



(a) 均方误差曲线　　(b) 均衡器输入



(c) MMA输出　　(d) WT-MMA输出

图 4.29 仿真结果

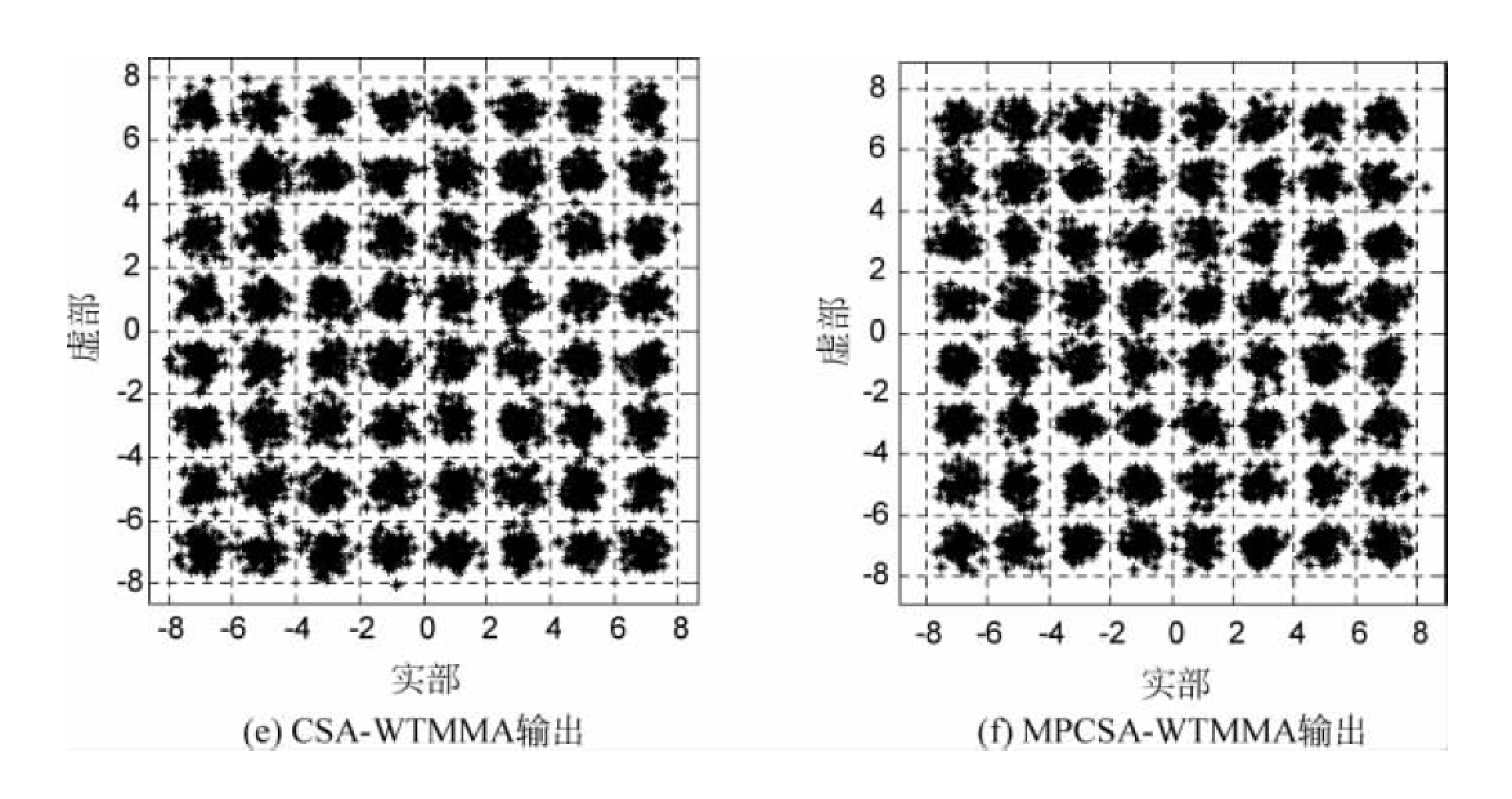



图 4.29 （续）

图 4.29(a)表明，在收敛速度上，MPCSA-WTMMA 与 CSA-WTMMA 基本相同，但比 MMA 快了近 9000 步、比 WTMMA 快了近 3500 步。在均方误差上，MPCSA-WTMMA 比 MMA 和 CSA-WTMMA 减小近 1.5dB，比 WTMMA 减小约 2.5dB。图 4.29(c)～(f)表明，MPCSA-WTMMA 的星座图比 MMA、WTMMA 和 CSA-WTMMA 更加清晰、紧凑。

## 参 考 文 献


[1] Shang R H，Jiao L C，Gong M G，et al. Clonal selection algorithms for dynamic multi-objective optimization[C]. Proceedings of the International Conference on Conference on Computational Inteligence and Security. Berlin：Springer-Verlag，2005，3801：846～851.

[2] 焦李成，尚荣华，马文萍等. 多目标优化免疫算法、理论和应用[M]. 北京：科学出版社，2010.

[3] Ma W P，Jiao L C，Gong M G. Immunodominance and clonal selection inspired multiobjective clustering[J]. Progress in Natural Science，2009，19(6)：751～758.

[4] 郑日荣，毛宗源，罗欣贤. 基于欧式距离和精英交叉的免疫算法研究[J]. 控制与决策，2005，20(2)：161～169.

[5] 王海英. 智能优化算法研究及其在移动机器人相关技术中的应用[D]. 哈尔滨：哈尔滨理工大学，2007.

[6] Jiao L C，Wang L. A novel genetic algorithm based on immunity[J]. IEEE Transaction on Systems，Man，and Cybernetics—Part A：Systems and Humans，2000，30(5)：552～561.

[7] 王磊. 免疫进化计算理论及应用[D]. 西安：西安电子科技大学，2001.

[8] 王磊，潘进，焦李成. 基于免疫策略的进化算法[J]. 自然科学进展，2000，10(5)：451～455.

[9] 葛红，毛宗源. 免疫算法几个参数的研究[J]. 华南理工大学学报(自然科学版)，2002，30(12)：15～18.

[10] Felipe Campelo，Frederico G Guimarães，Hajime Igarashi，et al. A clonal selection algorithm for optimization in electromagnetics[J]. IEEE Transactions on Magnetics，2005，41(5)：1736～1739.

[11] 丁锐. 免疫优化算法在正交小波盲均衡中的应用研究[D]. 淮南：安徽理工大学，2012.

[12] 刘锋，葛临东，吴业进，等. 一种基于实数编码遗传算法的常数模盲均衡算法[J]. 电讯技术，2008，10(48)：5～8.

[13] Santamriai I，Ibanez J，Vielva L，et al. Bind equalization of constant modulus signals via support vector regression[C]. Proceedings of IEEE International Conference on Acoustics，Speech，and Signal Processing，Hong Kong：IEEE publication，2003，(2)：737～740.

[14] Marcelino Lazaro，Jonathan Gonzalez-Olasola. Blind equalization using the IRWLS formulation of the Support Vector Machine[J]. Signal Processing，2009，7(89)：1265～1270.

[15] 丁锐，郭业才. 基于免疫克隆的正交小波变换盲均衡算法[J]. 计算机工程与设计，2011，32(9)：3158-3163.

[16] 张志超. 基于正交小波变换支持向量机盲均衡算法[D]. 南京：南京信息工程大学，2011.

[17] 季童莹，郭业才，高敏. 引入支持向量机的小波分数间隔盲均衡算法[J]. 声学技术，2011，30(2)：178-183.

[18] 郭业才，孙凤. 基于人工免疫网络的正交小波盲均衡算法[J]. 计算机工程，2012，38(7)：158-181.

[19] Wenjie Tian，Yu Geng，Jicheng Liu，et al. Support vector regression and immune clone selection algorithm for intelligent electronic circuit fault

diagnosis[C]. 2009 Pacific-Asia Conference on Circuits, Communications and System, 2009：297～300.

[20] 董春曦，饶鲜，杨绍全，等. 支持向量机参数选择方法研究[J]. 系统工程与电子技术，2004，26(8)：1117～1120.

[21] 杨欣伟. 基于支持向量机的商场客流统计方法研究[D]. 天津：河北工业大学，2009.

[22] 马驰，阮秋琦. 基于离散为粒子群优化算法的 SVM 参数选择[J]. 计算机技术与发展，2007，17(12)：20～23.

[23] 杨洁，郑宁，刘董，等. 基于遗传算法的 SVM 带权特征和模型参数优化[J]. 计算机仿真，2008，25(9)：115～118.

[24] 燕飞，秦世引. 一种基于模拟退火的支持向量机超参数优化算法[J]. 航天控制，2008，26(5)：7～12.

[25] 魏圆圆，唐超礼，黄友锐. 自适应克隆选择算法及其仿真研究[J]. 模式识别与人工智能，2009，2(22)：202～207.

[26] Yang J, Werner J J, Dumont G A. The multimodulus blind equalization and its generalized algorithm[J]. IEEE Journal On Sel. Areas in Commun, 2002, 20(5)：997～1015.

[27] Jenq-Tay Yuan, Tzu-Chao Lin. Equalization and carrier phase recovery of CMA and MMA in blind adaptive receivers[J]. IEEE Tran. on Signal Processing, 2010, 58(6)：3206～3217.

[28] 窦高奇，高俊. 适用于高阶 QAM 系统的多模盲均衡新算法[J]. 电子与信息学报，2008，2(30)：388～391.

[29] 王彬，葛临东，吴迪，等. 适用于高阶 QAM 信号的盲均衡算法[J]. 通信学报，2010，31(8A)：10～16.

[30] 霍亚娟，葛临东，王彬. 适用于十字型正交幅度调制信号的多模盲均衡算法[J]. 计算机应用，2009，29(9)：2578～2580.

[31] 邵怀宗，吴颖. 一种新的适用于高阶 QAM 系统的均衡算法[J]. 电波科学学报，2009，24(5)：961～965.

[32] Feng Liu, Hu-cheng An, Jia-ming Li, et al. Build equalization using $v$-support vector regressor for constant modulus signals[C]. 2008 International Joint Conference on Neural Networks(IJCNN 2008), IEEE, 2008：161～164.

[33] 李金明，赵俊渭，陆晶. 基于支持向量机的水声信道盲均衡算法研究[J]. 电声基础，2006，6：4～6.

[34] 邓乃扬，田英杰著. 支持向量机：理论、算法与拓展[M]. 北京：科学出版

社,2009.

[35] Guo Yecai, Hu Lingling . An orthogonal wavelet transform multi-modulus blind equalization algorithm based on variable step-size of exponential function [C]. 2011 International Conference on Aerospace Engineering and Information Technology,2011.

[36] Guo Yecai, Ding Rui. A Support vector machine blind equalization algorithm based on immune clone algorithm [C]. 2011 International Conference on Intelligent Computing and Information Science,2011.

[37] 司书宾,孙树栋,徐娅萍.求解Job-Shop调度问题的多种群双倍体免疫算法研究[J].西北工业大学学报,2007,1(25):27~31.

[38] 余建军,孙树栋,吴秀丽,等.四种改进免疫算法及其比较[J].系统工程,2005,2(24):106~112.

[39] 叶在福,单渊达.多种群遗传算法在电网扩展规划中应用的改进[J].电力系统及其自动化学报,1999,5-6(11):55~61.

[40] 邓长春,朱儒明,李咏霞,等.一种求解TSP问题的多种群并行遗传算法[J].计算机仿真,2008,9(25):187~190.

[41] 蔡良伟,张基宏,李霞.作业车间调度问题的多种群遗传算法[J].电子学报,2005,6(33):991~994.

[42] 张英杰,范朝冬.求解可满足性问题全部解的改进多种群克隆免疫算法[J].信息与控制,2011,1(40):34~38.

[43] 史峰,王辉,郁磊,等.MATLAB智能算法30个案例分析[M].北京:北京航空航天大学出版社,2011.

# 第 5 章　基于粒子群算法优化的盲均衡算法

**【内容导语】** 本章在阐述粒子群优化算法原理及参数设置的基础上，分析研究了免疫克隆粒子群优化算法、量子粒子群优化算法及动态粒子群优化算法。建立了完整的基于粒子群算法优化的正交小波常模盲均衡算法、基于免疫克隆粒子群算法优化的正交小波常模盲均衡算法、基于量子粒子群算法优化的正交小波常模多模和加权多模盲均衡算法、基于动态粒子群算法优化的正交小波多模盲均衡算法。这些算法的性能较优化前有较大提高。

## 5.1　粒子群优化算法

粒子群优化（particle swarm optimization，PSO）算法[1-6]是美国社会心理学家 J. Kennedy 和电气工程师 R. C. Eberhart 受人工生命（探索自然界

生物的群体行为)研究结果的启发,通过模拟鸟群觅食过程中的迁徙和群居行为,及个体之间信息的社会共享对整体进化的推动作用,提出的一种基于群体智能的全局随机搜索算法。1995 年 IEEE 国际神经网络学术会议正式发表了题为 *Particle Swarm Optimization* 的论文,标志着 PSO 算法诞生。与传统的种群进化算法相比,PSO 算法采用速度-位置模型,操作简单、容易实现,避免了其他进化算法对个体进行交叉、变异、选择等操作。

PSO 算法是一种基于迭代的优化工具,通过不断更新位置和速度进化到全局最优,容易实现、可调参数少、无需梯度信息且运行效率高。它利用粒子的自身经验并共享其他个体信息来加快进化速度,搜索全局最优解;通过线性调整惯性权重保持粒子的惯性运动,使搜索空间不断扩展,保证收敛到最优位置。目前,在求解噪声和动态环境下的优化问题、TSP 问题、多目标优化问题、神经网络及模式分类、模糊前向神经网络模式分类等方面都取得了满意的效果。

### 5.1.1 粒子群优化算法原理

PSO 算法具有进化计算和群体智能的特点,模拟鸟群觅食行为的过程,它的基本思想是通过群体中个体之间的协作与信息共享来寻找种群最优解。所有的粒子都有一个位置和速度决定它们飞行的距离和方向,被优化的函数决定它的适应度值。在解空间中,其他粒子的搜索轨迹由当前最优粒子的位置和速度向量来引导。

在 PSO 算法中,搜索空间中的任何一个粒子都可以被看作是优化问题的可行解。首先初始化一群由 $N$ 个粒子组成的粒子群,在 $D$ 维搜索空间中进行最优解搜索,粒子的位置向量和速度向量所对应的目标函数值为该粒子的适应度值,用它来评价粒子的搜索性能,并引导种群的搜索;利用速度向量决定粒子的飞行方向和速率。设第 $i$ 个粒子位置向量为 $\boldsymbol{x}_i=(x_{i1}, x_{i2},\cdots,x_{iD})$和速度向量为 $\boldsymbol{v}_i=(v_{i1},v_{i2},\cdots,v_{iD})$,其中 $x_{id}$ 和 $v_{id}$ 分别表示第 $i$ 个粒子第 $d$ 维位置和第 $i$ 个粒子第 $d$ 维速度,确定其初始值为$[-1,1]$内的

随机数。通常其他粒子将追随当前最优粒子在解空间中进行搜索，在每一次迭代中粒子都是跟踪两个极值（个体极值和全局极值）来更新自己的位置向量和速度向量。

粒子 $i$ 在寻优过程中记录它当前的个体极值 $\boldsymbol{p}_i=(p_{i1},p_{i2},\cdots,p_{iD})$（个体极值 $p_i$ 指个体所经历位置中计算得到的适应度最优值位置向量）和整个粒子群当前的全局极值 $\boldsymbol{p}_g=(p_{g1},p_{g2},\cdots,p_{gD})$（全局极值 $p_g$ 是指种群中所有粒子搜索到的适应度最优值位置向量）；迭代到 $t+1$ 次时，第 $i$ 个粒子第 $d$ 维速度和位置的更新公式为

$$v_{id}(t+1)=wv_{id}(t)+c_1r_1(p_{id}(t)-x_{id}(t))+c_2r_2(p_{gd}(t)-x_{id}(t)) \tag{5.1.1}$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \tag{5.1.2}$$

$$w=(w_{\max}-(w_{\max}-w_{\min})/N)t \tag{5.1.3}$$

式中，$i=1,2,\cdots,N$，$d=1,2,\cdots,D$；$x_{id}(t)$表示第 $t$ 次迭代时第 $i$ 个粒子第 $d$ 维位置；$v_{id}(t)$表示第 $t$ 次迭代时第 $i$ 个粒子第 $d$ 维速度；$p_{id}(t)$表示第 $t$ 次迭代时第 $i$ 个粒子第 $d$ 维个体极值；$p_{gd}(t)$表示第 $t$ 次迭代时第 $i$ 个粒子第 $d$ 维全局极值；$c_1$ 和 $c_2$ 为加速因子；$r_1$ 和 $r_2$ 为在[0,1]范围内变化的随机数；$M$ 为 PSO 算法的最大迭代次数；$w$ 为惯性权重，$w_{\max}$和 $w_{\min}$分别为最大的和最小的惯性权重，其中较大的惯性权重有利于在更大空间范围内进行搜索，而相对较小的惯性权重则可以保证粒子群体收敛到最优位置，所以线性调整惯性权重值可以加快收敛速度。因此，式(5.1.1)～式(5.1.3)组成基本 PSO 算法。

PSO 算法基本流程描述如下。

步骤 1：初始化粒子群。随机初始化粒子的初始位置向量 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{iD})$和初始速度向量 $\boldsymbol{v}_i=(v_{i1},v_{i2},\cdots,v_{iD})$，并初始化其他粒子群参数，如惯性权重、学习因子、种群规模、粒子最大速度等。

步骤 2：计算每个粒子的适应度值。

步骤 3：比较每个粒子相应的适应度值，选出粒子所经历的最优位置向量 $\boldsymbol{p}_i$(即根据用户选择最大或最小适应度值所对应的位置向量)，如果最大

(最小),则将其作为该粒子的个体极值。

步骤 4:对于每个粒子,将群体内所经历的适应度最大(最小)值与其历史适应度最优值进行比较,如果最大(最小),则将其作为当前整个种群的全局最优位置 $\boldsymbol{p}_g$。

步骤 5:根据式(5.1.1)～式(5.1.3)更新粒子的位置向量和速度向量。

步骤 6:达到 PSO 算法最大迭代次数,结束;否则转到步骤 2。

PSO 算法流程如图 5.1 所示。

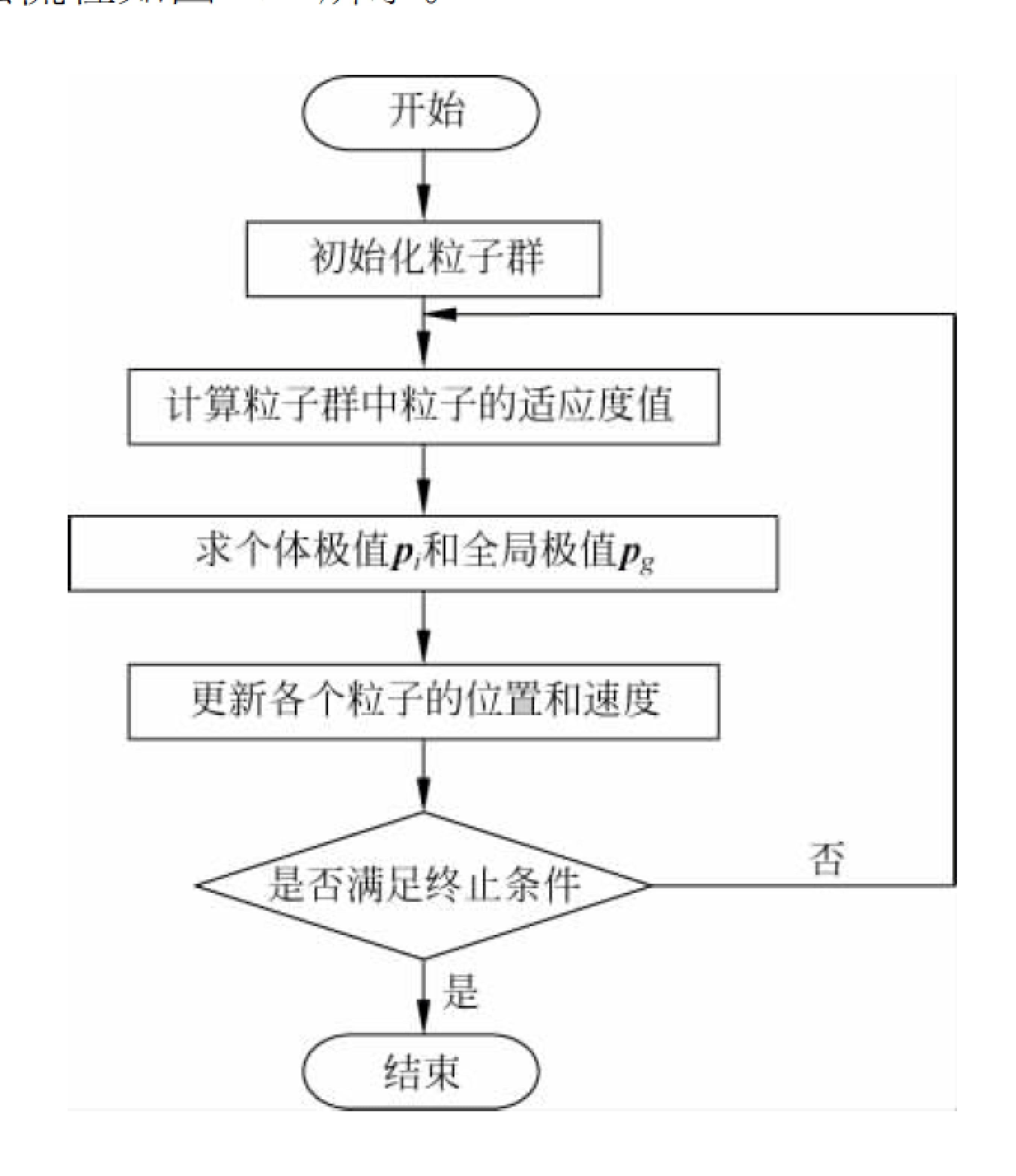



图 5.1 标准 PSO 算法流程图

### 5.1.2 粒子群优化算法参数设置

在 PSO 算法中,如何确定最佳参数使算法性能达到最优,是一个极其复杂的优化问题。各参数间具有相关性且参数空间大小不同,因此,在实际应用中,并没有通用的方法来确定算法的最佳参数,只能通过用户的经验选取。PSO 算法中主要参数有惯性权重、学习因子、种群规模、粒子最大速度和种群迭代次数等。

1. 惯性权重[7-8]

在PSO算法的可调参数中，惯性权重是最重要的参数，它是衡量粒子前一时刻的速度对下次移动的影响程度。也可以说，它是粒子对自己先前飞行速度的信任度，它对粒子搜索能力能够产生重要的影响。因此，PSO算法中全局搜索和局部搜索之间的平衡，可以通过调整惯性权重来实现。全局搜索能力强、局部搜索能力弱，设置较大的惯性权重；全局搜索能力弱、局部搜索能力强，设置较小的惯性权重。可见，适当选择惯性权重，能够提高PSO算法的性能。但对惯性权重的调整有一定的难度，当惯性权重较大时，有利于全局搜索且收敛速度快，但得不到精确解；当惯性权重较小时，有利于局部搜索并能得到解精确，但收敛速度变慢、易陷入局部极值。因此，要使算法的收敛速度和收敛精度维持平衡，可以通过适当的惯性权重来得到，这样可以减少运算量和迭代次数来得到最优解。

2. 学习因子

粒子本身经验和群体经验对粒子运动轨迹的影响主要是学习因子 $c_1$ 与 $c_2$，它反映了粒子间的信息交流。理想状态下，在搜索初期，要使粒子尽可能地搜索整个空间；在搜索末期，粒子应避免陷入局部极值。

$c_1$ 和 $c_2$ 分别结合随机向量 $r_1$ 和 $r_2$，$c_1$ 和 $r_1$ 控制着认知部分对于粒子速度的随机影响，$c_2$ 和 $r_2$ 控制着社会部分对于粒子速度的随机影响。当 $c_1=c_2$ 时，粒子将会在其现有速度的基础上，继续向上持续移动，碰到搜索边界时为止。如果 $c_1>0$ 且 $c_2=0$，所有粒子就是独立的爬山者，每个粒子都在其邻域内寻找新的更好的最优位置以替代当前的最优位置，这是局部搜索。反之，如果 $c_1=0$ 且 $c_2>0$，整个种群只存在一个吸引的位置点，所有粒子都会变成一个随机的爬山者选择自己的路线。

3. 种群规模

种群规模是指种群中粒子的个数。如何确定粒子的个数需依赖于具体

要解决的实际问题。

4. 粒子最大速度

粒子的初始速度向量具有随机性,通过粒子位置向量更新公式产生的运动轨迹也是不可预知的,为了抑制这种无规律的运动,将速度向量限制在$[-v_{max}, v_{max}]$。当$v_{max}$增大时,有利于粒子群的全局搜索;当$v_{max}$减小时,有利于粒子群的局部搜索。但$v_{max}$过大,粒子可能无规律地运动,超过最优解所在范围,使算法难以收敛而陷入停滞状态;相反,当$v_{max}$过小时,粒子运动步长太短时,算法又有可能陷入局部极值点。因此,$v_{max}$的选择通常根据用户的实际经验来确定。

5. 种群迭代次数

种群迭代次数也是依赖于具体问题来设计的,合适的迭代次数可以使算法得到一个最优解,迭代次数太小或太大可能使算法早熟或复杂度增大。因此,在PSO算法中是假设一定的迭代次数,作为停止准则。

### 5.1.3 改进的粒子群优化算法

PSO算法在早期收敛速度比较快,但收敛后精度较低、易发散。为了克服此不足,可对PSO算法进行改进。下面举例说明:其一,在PSO算法中,若惯性权重、最大速度等参数较大时,最后得到的优化解不一定是最优解,甚至算法不收敛;其二,假设种群收敛,所有的粒子都被最优解的方向所牵引,导致种群的多样性丢失,使种群迭代后期的收敛速度变慢,收敛到一定精度时无法继续进行优化,精度较低。

改进PSO算法性能主要从以下几个方面来进行:①提高粒子种群的多样性;②加快收敛速度;③算法停滞或陷入局部最优后的处理;④与其他智能优化算法结合。针对粒子群优化算法存在的缺点,给出如下改进算法:

(1) 免疫克隆粒子群优化算法,它是在粒子群体进化过程中,不断对粒子群体进行免疫克隆,根据生物群体中"优胜劣汰"的原理,提高种群的多样性;

(2) 量子粒子群优化算法,它是将量子理论应用于 PSO 算法中,避免了 PSO 算法中参数调节的复杂性,对更少的参数进行优化,加快了收敛速度、提高了稳定性;

(3) 动态粒子群优化算法,它是根据 PSO 算法迭代后期种群多样性变差,对粒子群体进行部分初始化,将新产生的部分粒子群体覆盖原来多样性变差的部分粒子群体,增强了种群的多样性。

以上的改进算法有效解决了局部搜索能力与全局搜索能力之间的平衡问题。

## 5.2 基于粒子群算法优化的正交小波常模盲均衡算法

本节主要研究了以 PSO 算法来初始化均衡器权向量,并结合小波变换理论,研究了基于粒子群算法优化的常模盲均衡算法(particle swarm optimization based constant modulus blind equalization algorithm, PSO-CMA)和基于粒子群算法优化的正交小波常模盲均衡算法(particle swarm optimization based orthonormal wavelet transform constant modulus blind equalization algorithm, PSO-WTCMA)[9-10]。

### 5.2.1 基于粒子群算法优化的常模盲均衡算法

PSO 算法是一种全局随机搜索智能算法,通过迭代不断更新粒子的位置和速度,以保证该算法收敛到粒子最优位置,避免陷入局部极值点。因此,将 PSO 算法引入到 CMA 中,利用 PSO 算法的全局搜索性能,寻找最优的初始化权向量,可以弥补 CMA 中利用代价函数更新权向量易陷入极值的缺陷。

### 1. PSO-CMA 描述

(1) 基本思想

随机初始化一组权向量,用这组权向量作为 PSO 算法的决策变量,将均衡器输入信号作为 PSO 算法的输入信号,并由 CMA 代价函数确定 PSO 算法的适应度函数,通过迭代寻找到适应度最优值对应的粒子位置向量,并作为 CMA 的初始化权向量,这就是基于粒子群算法优化的常模盲均衡算法(PSO-CMA),其原理如图 5.2 所示。

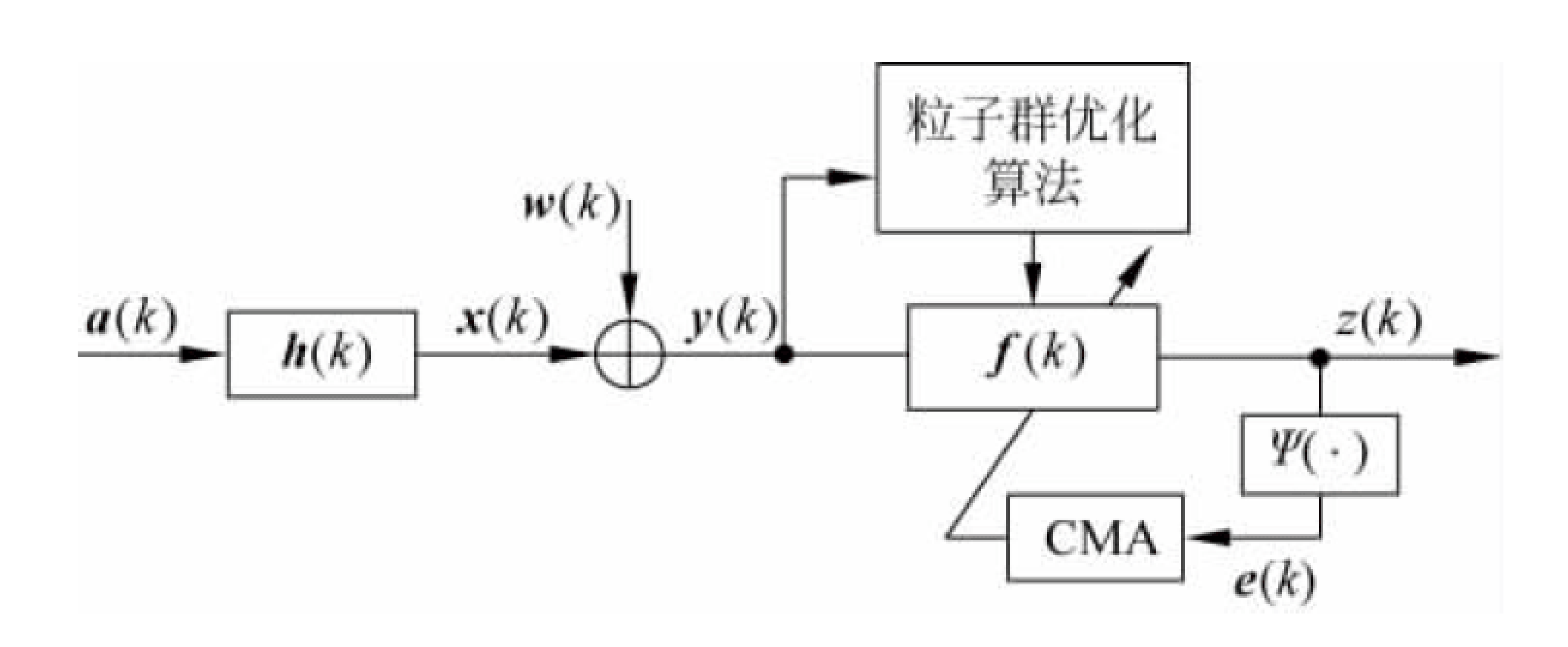



图 5.2 PSO-CMA 原理

图中,$\boldsymbol{a}(k)$是零均值独立同分布发射信号;$\boldsymbol{h}(k)$是信道冲激响应向量;$\boldsymbol{w}(k)$是加性高斯白噪声向量;$\boldsymbol{y}(k)$是均衡器的输入向量;$\boldsymbol{\Psi}(\cdot)$是误差生成函数,$e(k)$是误差函数;$\boldsymbol{f}(k)$是均衡器权向量;$z(k)$是均衡器的输出信号。

$$\boldsymbol{y}(k) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k) + \boldsymbol{w}(k) \tag{5.2.1}$$

均衡器的输出为

$$z(k) = \boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{f}(k) \tag{5.2.2}$$

均衡器的误差为

$$e(k) = R^2 - | z(k) |^2 \tag{5.2.3}$$

CMA 的代价函数为

$$J = E[e^2(k)] = E\{[z(k) - R^2]^2\} \tag{5.2.4}$$

权向量的迭代公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) - \mu z(k)(| z(k) |^2 - R_2)\boldsymbol{y}^*(k) \tag{5.2.5}$$

式中，$\mu$ 为步长，式(5.2.1)～式(5.2.5)就构成了 CMA，利用 PSO 算法搜索均衡器的最优权向量。

(2) 适应度函数的确定

CMA 代价函数为

$$J_{\mathrm{CMA}}(k) = (|z(k)|^2 - R^2)^2 \tag{5.2.6}$$

将 PSO 算法应用于盲均衡算法中需要构造合适的适应度函数。为此，将适应度函数定义为

$$\mathrm{Fit}(\boldsymbol{f}_i) = 1/J_{\mathrm{CMA}}(\boldsymbol{f}_i), \quad i = 1,2,\cdots,M \tag{5.2.7}$$

式中，$J_{\mathrm{CMA}}(\boldsymbol{f}_i)$是均衡器的代价函数，$\boldsymbol{f}_i$ 是 PSO 算法中粒子的位置向量，对应于均衡器的一个权向量。

(3) 最优权向量个体的选择

在 PSO 算法中，每个粒子都代表了极值优化问题中的一个潜在解，每个粒子的适应度值由适应度函数计算得到，而适应度函数是由 CMA 代价函数的倒数来确定。可见，求代价函数的最小值就是求 PSO 算法适应度函数的最大值，并获得此适应度值对应的最优位置向量。通过 PSO 算法的寻优迭代，并比较下一次迭代新粒子的适应度值和个体极值、全局极值的适应度值，更新个体极值和全局极值的位置向量，找到适应度最大值所对应的权向量个体(全局最优位置)，将此权向量作为 PSO-CMA 的初始化权向量。

(4) 迭代步长的选择

盲均衡算法实质是一种自适应滤波算法，通过利用随机信号的一些统计特性，从干扰信号中最佳地提取有用信号，根据自适应系统性能准则使均衡器的输出性能达到最优，而最优滤波准则规定了与某种特性信号的最佳参数。当盲均衡算法在其他条件一定的情况下，迭代步长就是控制算法收敛速度快慢，决定算法能否收敛于最优点的关键。在满足收敛条件的情况下，步长越小，收敛速度会减慢，但码间干扰较小；而步长越大，收敛速度会加快，但码间干扰较大。因此，算法在迭代步长的选择上利用了最小均方算法原理[11](least mean square，LMS)的加权向量收敛理论，选择适当的步长

因子，才能保证满足加权向量的收敛条件。通过以上分析，一个性能优良的盲均衡器应该是在算法收敛后，步长因子较小，以减小码间干扰，获得良好的均衡性能。

2. 算法仿真

以 CMA 为比较对象，与 PSO-CMA 进行比较，验证其有效性。

**【实验 5.1】** 两径水声信道为 $\boldsymbol{h}=[-0.35 \quad 0 \quad 0 \quad 1]$，均衡器权长为 16，信噪比为 20dB。当发射信号为 8PSK 时，$\mu_{\mathrm{CMA}}=0.0009$，$\mu_{\mathrm{PSO\text{-}CMA}}=0.0003$，第 4 个抽头系数设置为 1，其余为 0。当发射信号为 16QAM 时，$\mu_{\mathrm{CMA}}=0.000\,03$；$\mu_{\mathrm{PSO\text{-}CMA}}=0.000\,001$，第 6 个抽头系数设置为 1，其余为 0。在保证眼图完全清晰睁开的前提下，500 次蒙特卡罗仿真结果如图 5.3 所示。

图 5.3(a)表明，在收敛速度上，PSO-CMA 比 CMA 大约快了 2500 步；稳态误差上，PSO-CMA 与 CMA 相比，减小了近 2dB。图 5.3(b)表明，在收敛速度上，PSO-CMA 比 CMA 大约快了 1000 步；在稳态误差上，PSO-CMA 与 CMA 相比，减小了近 2.5dB。

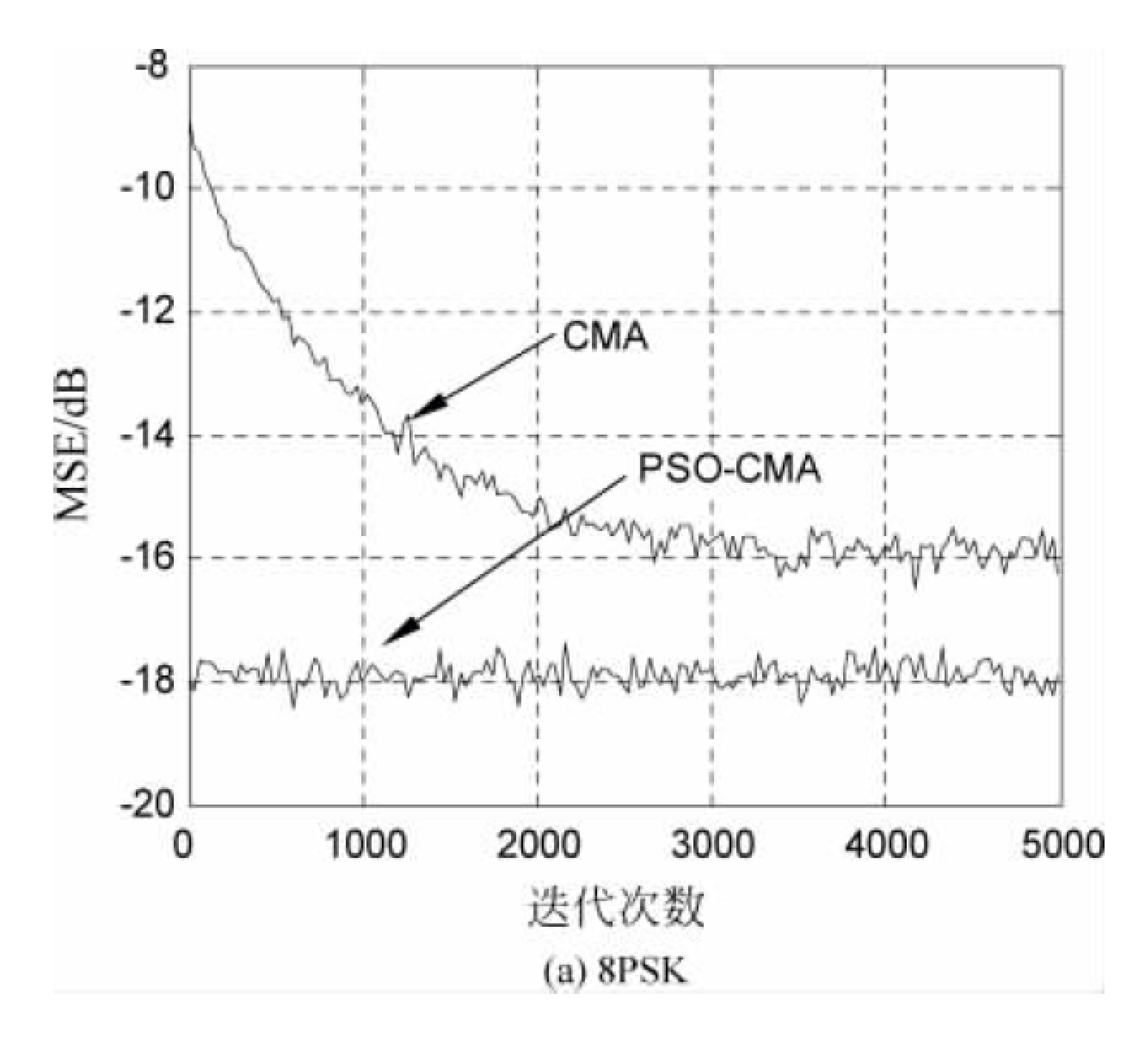



图 5.3 仿真结果

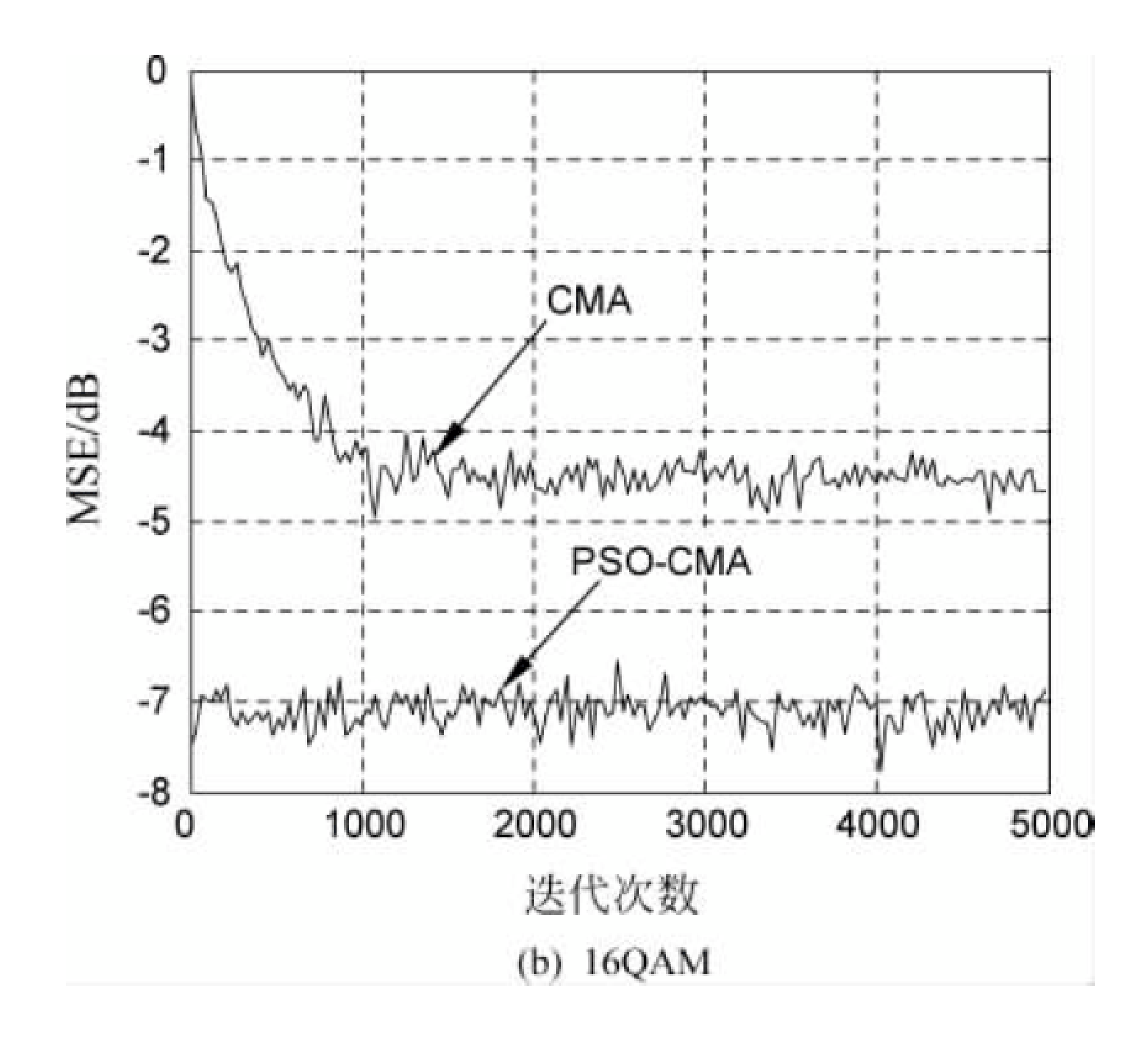



图 5.3 （续）

### 5.2.2 基于粒子群算法优化的正交小波常模盲均衡算法

文献[13～17]表明，在盲均衡算法中，影响盲均衡算法收敛速度的主要因素是输入信号的自相关矩阵。然而，利用小波变换理论来降低输入信号的自相关性，改善了 CMA 结构，在一定程度上可以加快收敛速度，性能稳健。本节结合 PSO 算法的全局搜索能力，研究了基于粒子群算法优化的正交小波常模盲均衡算法（PSO-WTCMA）。

#### 1. PSO-WTCMA 描述

PSO 算法通过迭代来寻找均衡器的最优权向量，而 WTCMA 的代价函数对均衡器权向量求梯度后，得到的寻优权向量迭代方程，缺乏全局搜索能力。通过 PSO 算法的寻优迭代，可以快速找到适应度最大值所对应的权向量个体（即粒子全局最优位置向量）并作为 PSO-WTCMA 的初始化权向量。

PSO-WTCMA 原理如图 5.4 所示。

由图 5.4，得

$$\boldsymbol{y}(k) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k) + \boldsymbol{w}(k) \tag{5.2.8}$$

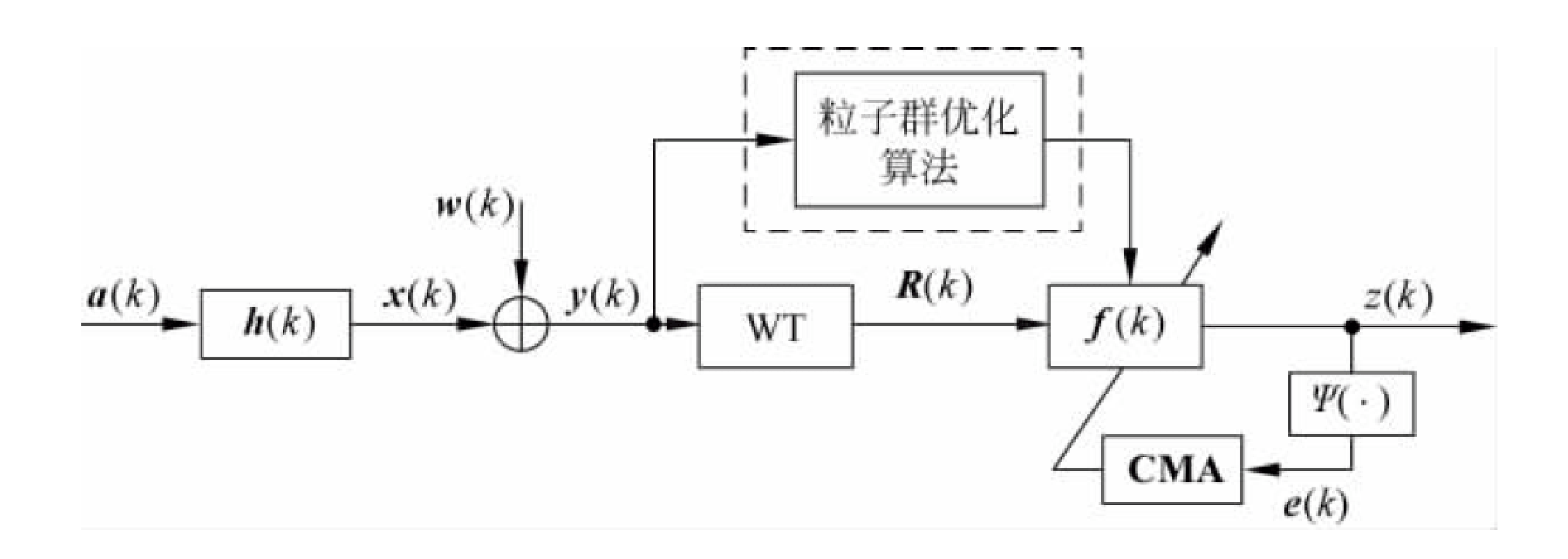



图 5.4　PSO-WTCMA 算法原理

$$\boldsymbol{R}(k)=\boldsymbol{y}(k)\boldsymbol{V} \tag{5.2.9}$$

$$z(k)=\boldsymbol{f}^{\mathrm{T}}(k)\boldsymbol{R}(k) \tag{5.2.10}$$

$$e(k)=R^2-|z(k)|^2 \tag{5.2.11}$$

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu\hat{\boldsymbol{R}}^{-1}(k)z(k)(|z(k)|^2-R^2)\boldsymbol{R}^*(k) \tag{5.2.12}$$

式中，$\boldsymbol{R}(k)$是 $\boldsymbol{y}(k)$经过正交小波变换后的信号向量；$\boldsymbol{V}$ 为正交小波变换矩阵；$\mu$ 为步长；$\hat{\boldsymbol{R}}^{-1}(k)=\mathrm{diag}[\sigma_{1,0}^2(k),\sigma_{1,1}^2(k),\cdots,\sigma_{I,k_j-1}^2(k),\sigma_{J+1,0}^2(k),\cdots,\sigma_{J+1,k_j-1}^2(k)]$，且 $\sigma_{i,n}^2(k)$与 $\sigma_{J+1,n}^2(k)$分别表示对小波变换系数 $u_{j,n}(k)$与尺度变换系数 $s_{J,n}(k)$的平均功率估计，其迭代公式为

$$\hat{\sigma}_{j,n}^2(k+1)=\beta\hat{\sigma}_{j,n}^2(k)+(1-\beta)\,|u_{j,n}(k)|^2 \tag{5.2.13a}$$

$$\hat{\sigma}_{J+1,n}^2(k+1)=\beta\hat{\sigma}_{J+1,n}^2(k)+(1-\beta)\,|s_{J,n}(k)|^2 \tag{5.2.13b}$$

式中，$\beta$ 是平滑因子，且 $0<\beta<1$。式(5.2.8)～式(5.2.13)就构成了正交小波变换常模盲均衡算法(WTCMA)。用 PSO 算法优化 WTCMA 就获得了 PSO-WTCMA。

## 2. 计算复杂度分析

设目标向量维数为 $D$，粒子群体为 $N$，种群最大迭代次数为 $M$，则初始化粒子群的计算复杂度为 $O(2ND)$，计算粒子群中每个粒子群适应度值的计算复杂度为 $O(N^2)$，种群更新的计算复杂度为 $O(2NM)$，因此，在一次迭代中，总计算复杂度为 $O(2ND+N^2+2NM)$，仅与粒子的规模和最大迭代

次数有关，计算量相对较小。在 $\boldsymbol{V}$ 已知的情况下，讨论权系数更新一次所需要的乘法次数为其计算复杂度，WTCMA 在每次权系数迭代过程中，需要运行 $L$ 点信号 $X(k)$（$L$ 为均衡器的长度）的正交小波变换，由小波分析理论可知，正交小波变换矩阵 $\boldsymbol{V}$ 为 $L\times L$，因此，计算式(5.2.9)得 WTCMA 的计算复杂度为 $O(L^2)$。实际上 $\boldsymbol{V}$ 为稀疏矩阵，假设 $\boldsymbol{V}$ 中每一行的非零元素数目为 $H$，这时计算式(5.2.9)的计算复杂度为 $O(HL)$；由式(5.2.9)、式(5.2.10)和式(5.2.13)知，WTCMA 完成一次权系数更新的计算复杂度为 $O(5L+HL)$。由式(5.2.5)知，CMA 完成一次权系数更新的计算复杂度为 $O(5L)$。可见，PSO-WTCMA 的计算复杂度为 $O(2ND+N^2+2NM+5L+HL)$。显然，PSO-WTCMA 的计算量增加，主要是 PSO 算法引起的，由于粒子群规模通常较小，故所增加的计算量不是太大。

3. 算法仿真

为了检验 PSO-WTCMA 的有效性，以 CMA、PSO-CMA 和 WTCMA 为比较进行对较，进行仿真实验。

**【实验 5.2】** 混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；发射信号为 8PSK，均衡器权长为 16，信噪比为 20dB，$\mu_{\text{CMA}}=0.0009$、$\mu_{\text{WTCMA}}=0.0025$、$\mu_{\text{PSO-CMA}}=0.000\,22$、$\mu_{\text{PSO-WTCMA}}=0.0005$；在 CMA 和 WT-CMA 中，第 4 个抽头系数设置为 1，其余为 0；信道输入信号采用 DB2 正交小波分解，分解层数为 2，遗忘因子 $\beta=0.999$，功率初始值设置为 4。

两径水声信道 $\boldsymbol{h}=[-0.35\quad 0\quad 0\quad 1]$；发射信号为 16QAM，均衡器权长为 16，信噪比为 20dB；$\mu_{\text{CMA}}=0.000\,01$、$\mu_{\text{WTCMA}}=0.0002$、$\mu_{\text{PSO-CMA}}=0.000\,04$、$\mu_{\text{PSO-WTCMA}}=0.000\,25$；在 CMA 和 WTCMA 中，第 6 个抽头系数设置为 1，其余为 0；信道输入信号采用 DB2 正交小波分解，分解层数为 2，遗忘因子 $\beta=0.9999$，功率初始值设置为 4。

在眼图完全清晰睁开的前提下，20 次蒙特卡罗仿真结果如图 5.5 所示。

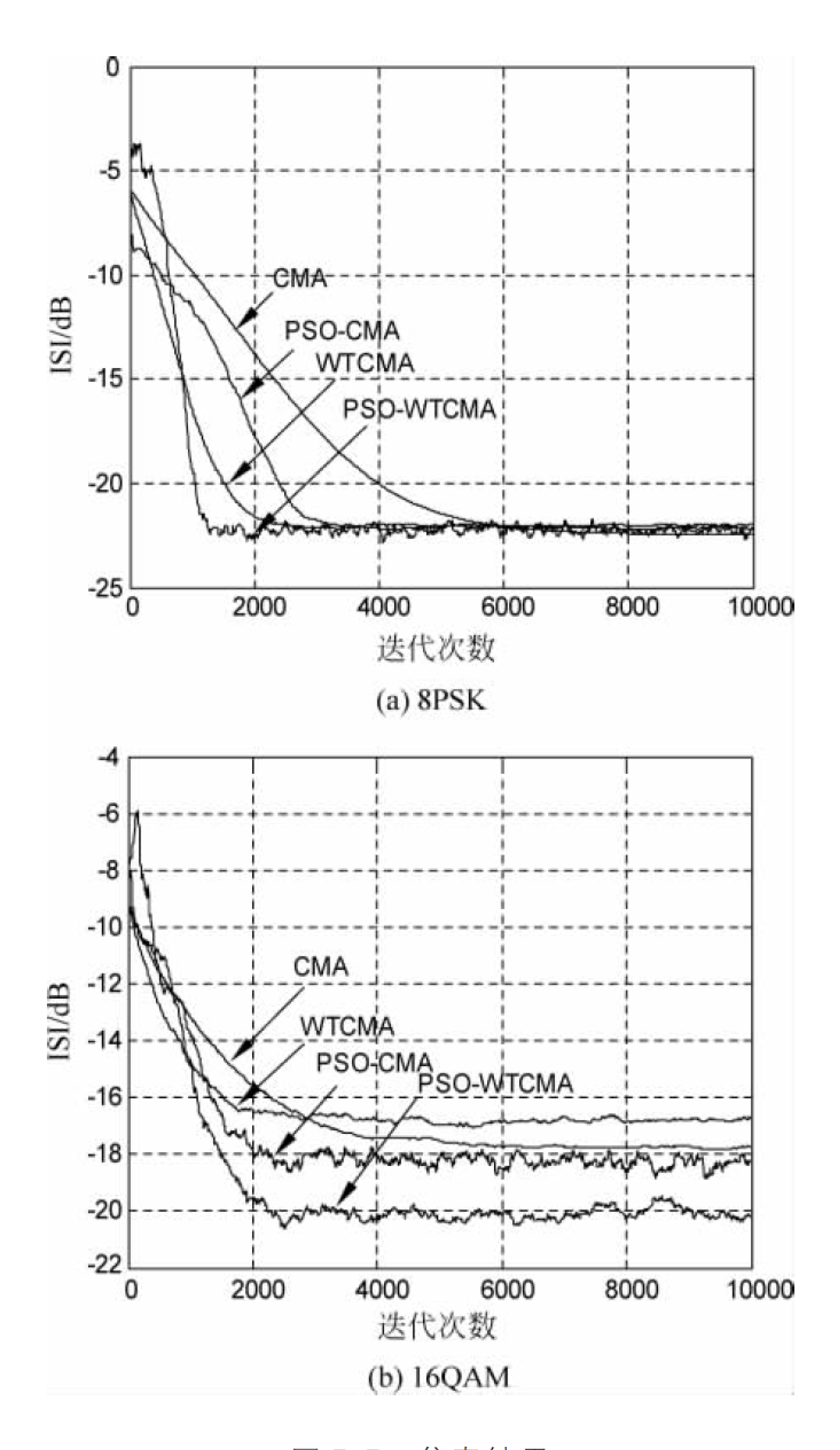



图 5.5 仿真结果

图 5.5(a)表明，PSO-WTCMA 的收敛速度最快，比 CMA 快约 5000 步、比 WTCMA 快约 2500 步、比 PSO-CMA 快约 1500 步；这几种算法的收敛后，剩余码间干扰近似相同，约达到－22dB。图 5.5(b)表明，PSO-WTCMA 的收敛速度最快，比 CMA 快约 4500 步、比 WTCMA 快约 2500 步、比 PSO-CMA 快约 1500 步；这几种算法的收敛后，PSO-WTCMA 的剩余码间干扰最小且达到了－20dB，比 CMA 约小 2dB、比 PSO-CMA 约小 1.8dB、比 WTCMA 约小 3.2dB。

## 5.3　基于免疫克隆粒子群算法优化的正交小波常模盲均衡算法

本节在分析小波变换理论、免疫克隆(immune clone,IC)算法和PSO算法的基础上,研究了一种基于免疫克隆粒子群算法优化的正交小波常模盲均衡算法(immune clone particle swarm algorithm based WTCMA,ICPS-WTCMA)[18]。该算法将IC算法融合到PSO算法中,充分发挥了PSO算法的随机全局搜索能力和IC算法可以有效地避免种群陷入局部极值的优点,并由小波变换理论对输入信号进行去相关。

### 5.3.1　ICPS-WTCMA思想

随机初始化粒子群体,设粒子的位置向量为初始化权向量,一个粒子对应一组权向量,用这些权向量作为免疫克隆粒子群优化算法(immune clone particle swarm,ICPS)的决策变量,将均衡器输入信号作为ICPS的输入,ICPS的适应度函数定义为均衡器代价函数的倒数。以PSO算法的运算流程作为主体流程,将IC算法融入其中,以调整及优化粒子群体,通过迭代找到适应度最大值所对应的位置向量并作为WTCMA的初始化权向量。因此,将ICPS引入到WTCMA中,利用ICPS优化均衡器权向量,以此减小WTCMA局部收敛的可能性、降低稳态误差。ICPS-WTCMA原理如图5.6所示。

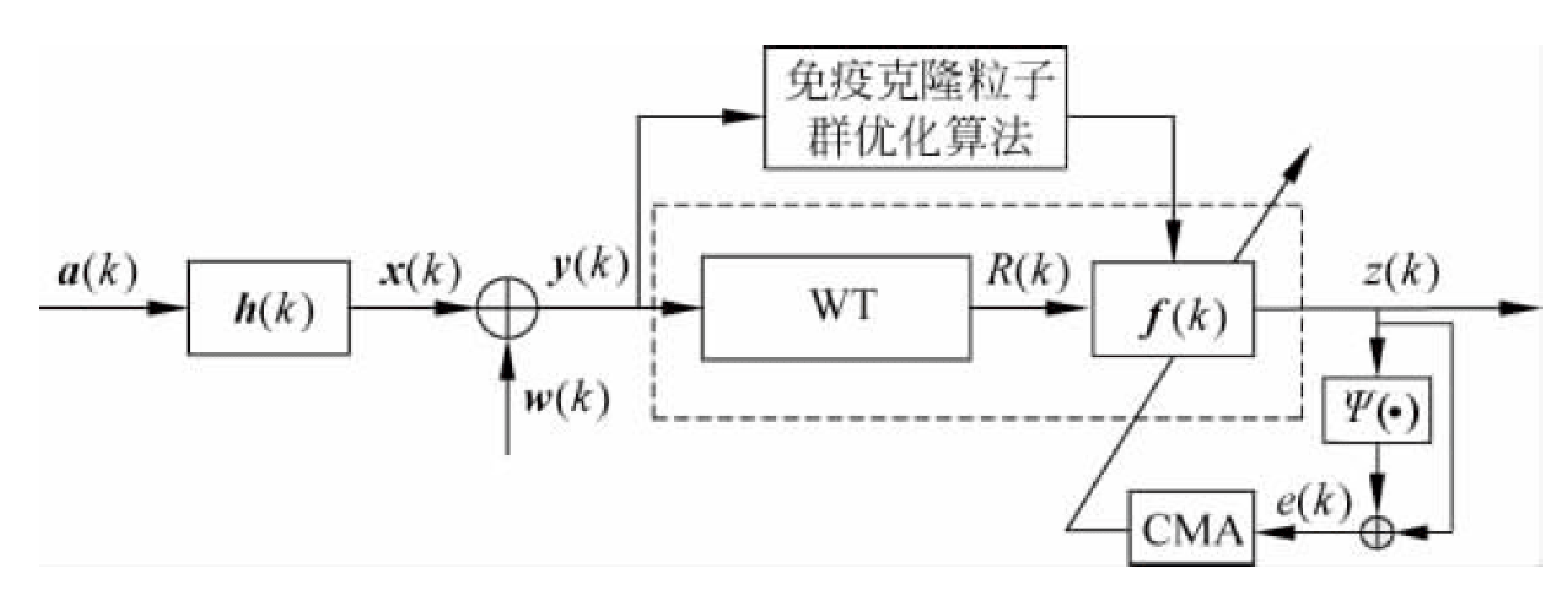



图5.6　ICPS-WTCMA原理

### 5.3.2 ICPS-WTCMA 描述

在多维的目标搜索空间中，PSO 算法与其他的全局优化算法同样存在着早熟收敛现象。在算法迭代期间引入免疫克隆机制，对避免早熟收敛现象具有一定的作用。IC 算法根据抗体亲和度值大小进行克隆，产生变异解群体，有助于防止早熟并扩大搜索范围。因此，将 IC 算法引入到 PSO 算法中，确定性地把进化群体中最好的候选解选择出来，参与 PSO 算法进化，增强了粒子种群的多样性，有效地改善了 PSO 算法在搜索后期由于种群多样性变差而引起的寻优效率低和早熟，提高了搜索速度。因此，将具有全局搜索特点和避免种群多样性变差的 IC 算法中引入到 PSO 算法中，可以弥补 PSO 算法易陷入局部极值的不足。具体思路：将粒子群看作一个免疫系统[19-21]，在 $D$ 维搜索空间中粒子的个数为 $N$，首先随机初始化粒子（即初始化抗体）的速度向量和位置向量，并设粒子的位置向量为它的决策向量。

设随机产生的一个初始种群为 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N]$，其中 $\boldsymbol{f}_i$ 是第 $i$ 个个体的位置向量对应于均衡器一个权向量，$0<i\leqslant N$。将粒子作为抗体，最优解作为抗原，用亲和度表示抗体与抗原之间的接近程度，选定 PSO 算法的适应度函数作为抗体的亲和度（即为待优化的目标函数），算法迭代一次后，将选择出来的抗体（粒子）亲和度（适应度）较高的抗体进行免疫克隆操作，通过对抗体群（粒子群）进行免疫克隆操作，不断产生新一代的抗体群（粒子群），免疫克隆实际上包括选择、克隆、高频变异和再选择，然后将免疫克隆后的最优解继续进行 PSO 算法的迭代。每次迭代过程中将选出的最优解进行免疫克隆操作，迭代结束后选出最优的位置向量，并将此位置向量作为均衡器的初始化权向量进行迭代。

与 CMA 的代价函数一样，WTCMA 的代价函数为

$$J_{\mathrm{WTCMA}}=(|z(k)|^2-R^2)^2 \tag{5.3.1}$$

PSO 算法是通过迭代找到适应度最优值，而 IC 算法是将选出抗体亲

和度最高的进行粒子群优化迭代，所以将 PSO 算法的适应度函数作为抗体的亲和度函数来进行优化。所以

$$\mathrm{Fit}(\boldsymbol{f}_i) = 1/J_{\mathrm{WTCMA}}(\boldsymbol{f}_i), \quad i = 1,2,\cdots,M \tag{5.3.2}$$

将式(5.3.2)作为 ICPS 算法的适应度函数。式中，$J_{\mathrm{WTCMA}}(\boldsymbol{f}_i)$是均衡器的代价函数，$\boldsymbol{f}_i$ 是粒子的位置向量对应于均衡器的一个权向量。

ICPS-WTCMA 充分利用 PSO 算法和 IC 算法的优点，来优化 WTCMA 的权向量。ICPS 算法具有以下特点：

(1) 保持了 PSO 算法的收敛速度快和信息共享的优点。

(2) 免疫克隆操作主要包括克隆、免疫基因和选择操作。克隆操作扩大了搜索空间；免疫基因操作包括了交叉和变异，提高种群的多样性主要靠高频变异来实现；选择操作使得优秀个体的基因得到继承，提高了算法的寻优能力、避免了退化。

### 5.3.3　ICPS-WTCMA 实施步骤

ICPS-WTCMA 的具体实现步骤如下。

步骤 1：在 $D$ 维空间中随机产生 $N$ 个粒子即抗体，初始化系统中各种参数。

步骤 2：计算当前每个粒子(抗体)的适应度值(亲和度值)。

步骤 3：克隆操作是根据个体亲和度(适应度)值大小对种群中每个个体单独执行，按克隆复制比例进行克隆操作。选出 $Y$ 个适应度值较大的粒子进行克隆操作，克隆后的种群规模为 $X$。

步骤 4：对克隆群体 $X$ 进行免疫基因操作，即交叉和高频变异操作，生成新的粒子种群 $X'$，重新计算种群 $X'$的适应度值。

步骤 5：对种群 $X'$按照适应度值大小进行再选择操作，得到新的粒子种群 $Y'$，这时 $Y'=Y$。粒子种群的更新，如果当前群体中选出的适应度最大值对应的粒子位置小于经过免疫克隆选出的适应度最大值对应的粒子位置，则用免疫克隆后的粒子位置来代替初始种群的粒子位置，以此来继承

优良的基因，以保持种群的多样性，并计算出当前种群中的个体极值 $p_{pbest}$ 和全局极值 $p_{gbest}$。

步骤 6：按照种群迭代次数继续更新粒子群体的速度和位置向量，重新计算粒子的适应度值，然后进行免疫克隆操作，比较并更新个体极值 $p_{pbest}$ 和全局极值 $p_{gbest}$。粒子速度和位置向量的更新公式为

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \tag{5.3.3}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{5.3.4}$$

$$w = (w_{\max} - (w_{\max} - w_{\min})/N)t \tag{5.3.5}$$

式中，$i=1,2,\cdots,N$，$d=1,2,\cdots,D$；$x_{id}(t)$表示第 $t$ 次迭代时第 $i$ 粒子第 $d$ 维位置，$v_{id}(t)$表示第 $t$ 次迭代时第 $i$ 粒子第 $d$ 维速度，$p_{id}(t)$表示第 $t$ 次迭代时第 $i$ 粒子第 $d$ 维的个体极值，$p_{gd}(t)$表示第 $t$ 次迭代时第 $i$ 粒子第 $d$ 维全局极值；$c_1$ 和 $c_2$ 为加速因子，$r_1$ 和 $r_2$ 为在[0,1]范围内变化的随机数，$M$ 为粒子群算法的最大迭代次数；$w$ 为惯性权重，$w_{\max}$和 $w_{\min}$分别为最大的和最小的惯性权重，经过迭代 $M$ 次后，寻找到种群中粒子的最优位置向量，即均衡器最优权向量。

步骤 7：终止操作。如果已达到最大迭代次数，则选出全局最优位置向量(均衡器的初始化权向量)，即 $\boldsymbol{p}_{gbest}=(p_{g1},p_{g2},\cdots,p_{gD})$；否则转步骤 2。最后，将输出的粒子的最优位置向量作为盲均衡器的初始化权向量。

### 5.3.4 算法仿真

为了检验 ICPS-WTCMA 的有效性，以 WTCMA 和 PSO-WTCMA 为比较对象，进行仿真实验。

**【实验 5.3】** 当混合相位水声信道 $\boldsymbol{h}=[0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$时，发射信号为 8PSK，均衡器权长为 16，信噪比为 20dB；种群规模为 100，克隆复制控制因子 $p_s=2$，最优交叉概率为 0.2，变异概率为 0.1，最

大进化代数为 500；WTCMA 的第 10 个抽头初始化为 1。其他参数设置如表 5.1 所示。在保证眼图完全清晰睁开的前提下，1000 次蒙特卡罗仿真结果如图 5.7(a)所示。

表 5.1　仿真参数设置

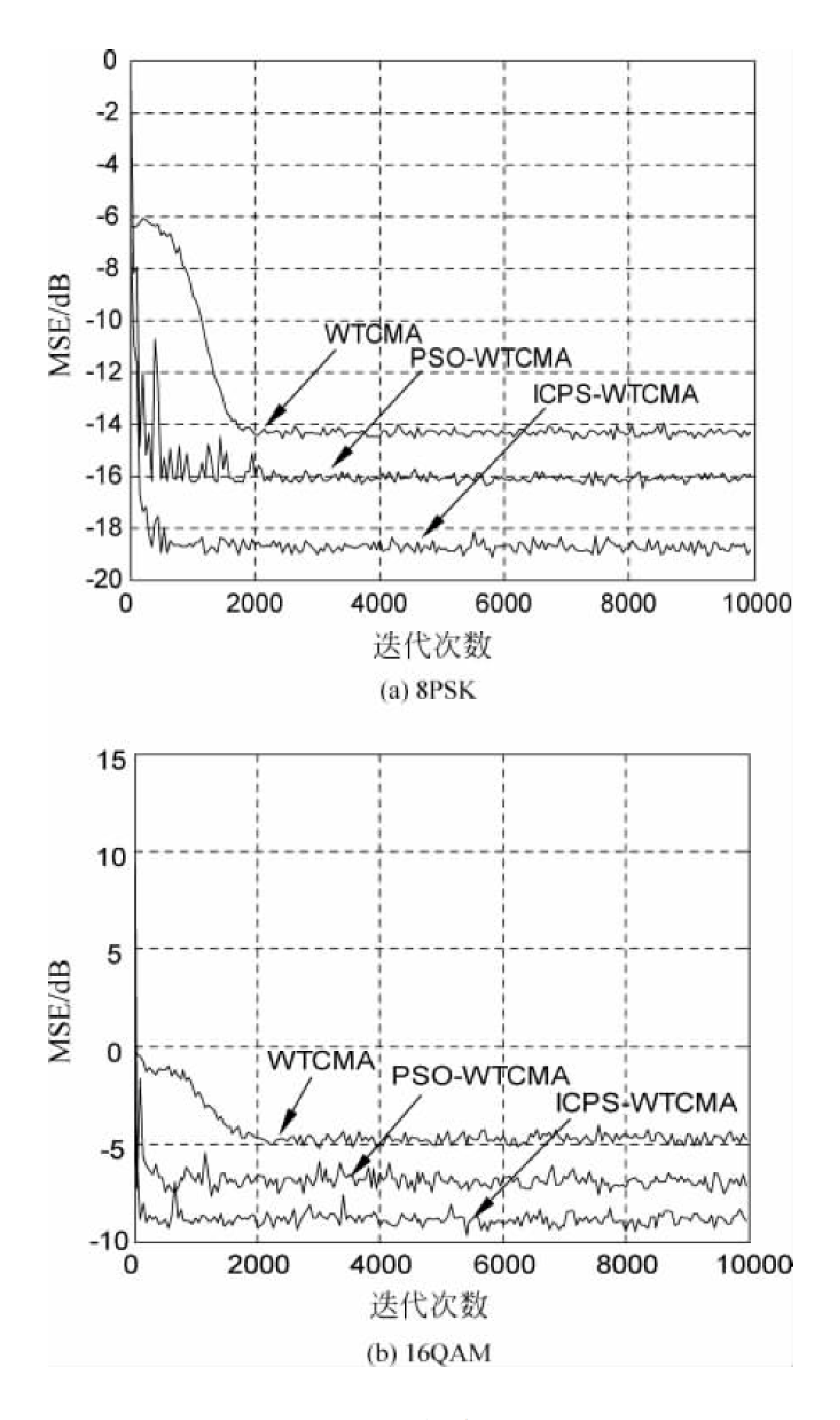
| 算　　法 | 仿真步长 | 小波 | 分解层数 | 初始化功率 | $\beta$ 取值 |
|---|---|---|---|---|---|
| WTCMA | 0.004 | DB4 | 2 | 4 | 0.95 |
| PSO-WTCMA | 0.000 15 | DB4 | 2 | 8 | 0.999 |
| ICPS-WTCMA | 0.000 18 | DB4 | 2 | 4 | 0.99 |

当最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$ 时，发射信号为 16QAM 信号，信噪比为 20dB，均衡器权长为 16，功率初始值设置为 4；种群规模为 100，克隆复制控制因子 $p_s=0.8$，最优交叉概率为 0.2，变异概率为 0.1，最大进化代数为 500；WTCMA 的第 8 个抽头初始化为 1。其他参数设置如表 5.2 所示。在保证眼图完全清晰睁开的前提下，300 次蒙特卡罗仿真结果如图 5.7(b)所示。

表 5.2　仿真参数设置

| 算　　法 | 仿真步长 | 小波 | 分解层数 | 初始化功率 | $\beta$ 取值 |
|---|---|---|---|---|---|
| WTCMA | 0.0006 | DB4 | 2 | 6 | 0.999 |
| PSO-WTCMA | 0.000 015 | DB4 | 2 | 6 | 0.999 |
| ICPS-WTCMA | 0.000 028 | DB4 | 2 | 10 | 0.99 |

图 5.7(a)表明，在收敛速度上，ICPS-WTCMA 比 PSO-WTCMA 大约快了 700 步，比 WTCMA 快了 1600 步；在稳态误差上，ICPS-WTCMA 与 PSO-WTCMA 相比，减小了近 2.5dB，与 WTCMA 相比，减小了近 6dB。图 5.7(b)表明，收敛速度上，ICPS-WTCMA 比 PSO-WTCMA 大约快了 200 步，比 WTCMA 快了 1400 步；稳态误差上，ICPS-WTCMA 与 PSO-WTCMA 相比，减小了近 1.5dB，与 WTCMA 相比，减小了近 3.5dB。



(a) 8PSK


(b) 16QAM

图 5.7　仿真结果

## 5.4　基于量子粒子群算法优化的正交小波常模盲均衡算法

在盲均衡算法中，CMA 利用接收信号的统计特性来均衡信号的。其中，基于分数间隔的常数模盲均衡算法(fractionally spaced blind equalization constant modulus algorithm，FSE-CMA)，在接收端以大于发送码元间隔 $T$

的速率对接收信号进行采样，避免了由于欠采样而引起的频谱混叠，提高了CMA性能。量子粒子群优化（quantum particle swarm optimization，QPSO）算法是一种具有量子行为的、以概率1搜索到全局最优解的智能优化算法，而PSO算法不能保证以概率1搜索到全局最优解[22]。本节将正交小波变换理论、FSE-CMA、多模盲均衡算法（multi-modulus blind equalization algorithm，MMA）与QPSO算法相结合，用QPSO算法优化均衡器权向量，以进一步改善盲均衡性能。

### 5.4.1 基于量子粒子群算法优化的正交小波分数间隔常模盲均衡算法

针对FSE-CMA[23-24]收敛速度慢、稳态误差大的缺点，通过分析QPSO算法和正交小波变换理论，研究了基于量子粒子群算法优化的正交小波分数间隔常模盲均衡算法（quantum particle swarm optimization algorithm based T/P fractionally spaced blind equalization wavelet transform constant modulus algorithm，QPSO-T/P-FSE-WTCMA）[9,25]。

#### 1. QPSO-T/P-FSE-WTCMA描述

在量子空间中，粒子满足聚集态的性质，所以QPSO算法是从量子力学的角度来研究粒子的收敛行为，它可以在整个可行解空间中进行全局优化搜索，其搜索性能远远优于PSO算法，能克服PSO算法在收敛性能上的缺陷。QPSO算法可调参数少、容易控制、进化方程简单。

QPSO-T/P-FSE-WTCMA的基本思想：随机初始化一组量子粒子的位置向量（初始化权向量），并把这些向量作为QPSO算法的决策变量，将各个子信道均衡器的输入信号作为QPSO算法的输入信号，并结合FSE-CMA的代价函数，确定QPSO算法的适应度函数，通过迭代寻找粒子的最优位置向量，并作为FSE-WTCMA中各个子信道均衡器中的初始化权向量。

QPSO-T/P-FSE-WTCMA 的原理结构如图 5.8 所示。

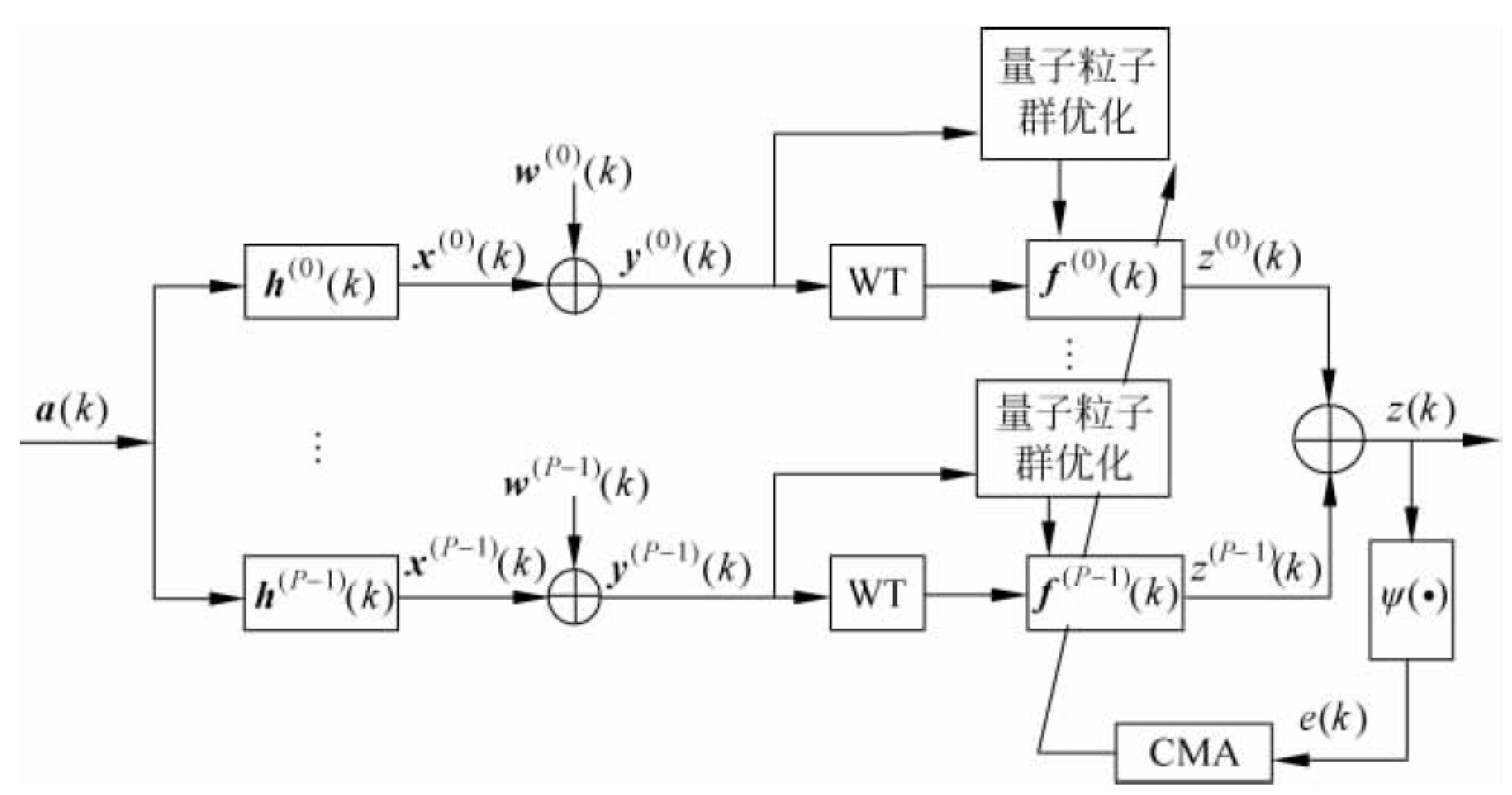



图 5.8 QPSO-T/P-FSE-WTCMA 原理

QPS 算法优化步骤如下。

步骤 1：初始化种群。随机初始化量子粒子的位置向量，并设置 QPSO 的空间解为[0,100]；随机产生初始种群为 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$，其中的一个个体 $\boldsymbol{f}_i(0<i\leqslant M)$对应于均衡器的一个权向量。

步骤 2：确定适应度函数。

分数间隔常模盲均衡器的代价函数为

$$J_{\text{FSE-CMA}}=(|z(k)|^2-R^2)^2 \tag{5.4.1}$$

QPSO 的适应度函数定义为

$$\text{Fit}(f_i)=1/J_{\text{FSE-CMA}}(f_i),\quad i=1,2,\cdots,M \tag{5.4.2}$$

式中，$J_{\text{FSE-CMA}}(f_i)$是 FSE-CMA 的代价函数，$f_i$ 是粒子的位置向量即均衡器权向量个体。

最优权向量的选择是将迭代后寻找到的位置向量，即 $\boldsymbol{p}_{\text{gbest}}=(p_{g1},p_{g2},\cdots,p_{gD})$作为 FSE-CMA 的初始化权向量。

2. QPSO 原理

QPSO-T/P-FSE-WTCMA 充分利用 QPSO 算法的全局优化搜索的优

点，来优化 FSE-WTCMA 的权向量。假设 PSO 系统是一个量子系统，粒子的位置向量和速度向量这两个信息不能同时确定，所以将其归结为一个参数：位置向量。QPSO 算法与 PSO 算法相比，具有以下优点：

(1) 量子系统是一个非线性的复杂系统，符合状态叠加原理，具有比一个线性系统更多的状态；在解空间中，可以更好地寻找最优位置向量。

(2) 量子系统没有既定的轨道，是一个不确定性的系统，具有广阔的搜索空间，可以以某一个概率出现在搜索空间中的任意一个位置。

(3) 在 PSO 算法中，粒子在一个有限的区域中，为了保持粒子群的聚集性，在束缚状态下进行寻找最优解；而在 QPSO 算法中，粒子能够以某一个概率出现在解空间中的任意一个位置，甚至可以出现在远离粒子自身所要收敛的随机点 $p$，或许这个位置可能比当前种群所找到的最优位置点 $p_{\text{best}}$ 具有更合适的适应度值。

QPSO 算法优化均衡器权向量步骤如下。

步骤 1：初始化量子粒子群参数。量子粒子的个数 $N$，量子粒子群的维数 $D$，在限定范围内，随机初始化量子粒子的位置向量 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{iD})$。

步骤 2：计算适应度值。根据式(5.4.2)计算量子粒子的适应度值。

步骤 3：更新量子粒子位置。更新每个量子粒子的新局部最优位置 $p_{id}$。

步骤 4：更新全局最优位置 $p_{gd}$。

步骤 5：计算量子粒子群中最优位置 $p_{\text{best}}$：在 QPSO 算法中，每个量子粒子必须收敛于各自的位置点 $\boldsymbol{x}=(x_1,x_2,\cdots,x_D)$，其中 $x_D$ 是量子粒子第 $D$ 维位置。按式(5.4.3)计算量子粒子的最优中值位置，$p_i$ 是量子粒子 $i$ 的最优位置。

$$m_{\text{best}}=(1/N)\sum_{i=1}^{N}p_i=\left[(1/N)\sum_{i=1}^{N}p_{i1},\cdots,(1/N)\sum_{i=1}^{N}p_{iD}\right] \tag{5.4.3}$$

步骤 6：按式(5.4.4)计算每个量子粒子的随机点 $p_i$，rand 为[0,1]内的随机数。

$$p_i=\text{rand}\cdot p_{id}+(1-\text{rand})p_{gd} \tag{5.4.4}$$

步骤7：按式(5.4.5)与式(5.4.6)更新每个量子粒子的位置，即

$$x(t+1)=[p\pm\beta\,|\,m_{\text{best}}-x(t)]\,|\,\ln(1/\text{rand}) \tag{5.4.5}$$

式中，$x(t)$为粒子的当前位置；$x(t+1)$为更新后的新位置；$\beta$为QPSO的创造力系数，它是QPSO算法收敛的一个重要参数，本节算法中采用的创造力系数公式为

$$\beta=(0.5-0.3M)*(M-t)/M+0.3 \tag{5.4.6}$$

式中，$M$为最大迭代次数，$t$表示迭代次数。

重复计算步骤2～步骤7，直到达到最大的迭代次数为止。

### 3. 算法仿真

现以T/2分数间隔盲均衡算法为基础，检验QPSO-T/2-FSE-WTCMA的有效性，以PSO-T/2-FSE-WTCMA、T/2-FSE-WTCMA、T/2-FSE-CMA和BSE-CMA(波特间隔盲均衡算法)为比较对象，进行仿真实验。仿真实验中，种群规模为40，最大进化代数为500，权向量为中心抽头初始化。

**【实验5.4】** 混合相位水声信道$\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；发射信号为8PSK，均衡器权长为32，信噪比为20dB。其余仿真参数如表5.3所示。

表5.3　仿真参数

| 算　　法 | 步长 | 小波 | 级数 | 初始化功率 | $\beta$取值 |
|---|---|---|---|---|---|
| BSE-CMA | 0.000 34 | | | | |
| T/2-FSE-CMA | 0.000 15 | | | | |
| T/2-FSE-WTCMA | 0.006 | DB2 | 3 | 4 | 0.9999 |
| PSO-T/2-FSE-WTCMA | 0.000 15 | DB2 | 3 | 4 | 0.95 |
| QPSO-T/2-FSE-WTCMA | 0.0045 | DB2 | 3 | 4 | 0.99 |

当发射信号为16QAM时，最小相位水声信道$\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$；信噪比为20dB；均衡器权长为32。其他仿真参数如表5.4所示。

表 5.4　仿真参数

| 算　　法 | 步长 | 小波 | 级数 | 初始化功率 | $\beta$取值 |
|---|---|---|---|---|---|
| BSE-CMA | 0.000 003 | | | | |
| T/2-FSE-CMA | 0.000 07 | | | | |
| T/2-FSE-WTCMA | 0.000 12 | DB2 | 2 | 4 | 0.999 |
| PSO-T/2-FSE-WTCMA | 0.000 05 | DB2 | 2 | 4 | 0.99 |
| QPSO-T/2-FSE-WTCMA | 0.000 012 | DB2 | 2 | 4 | 0.999 |

在保证眼图完全清晰睁开的前提下，仿真结果如图 5.9 所示。

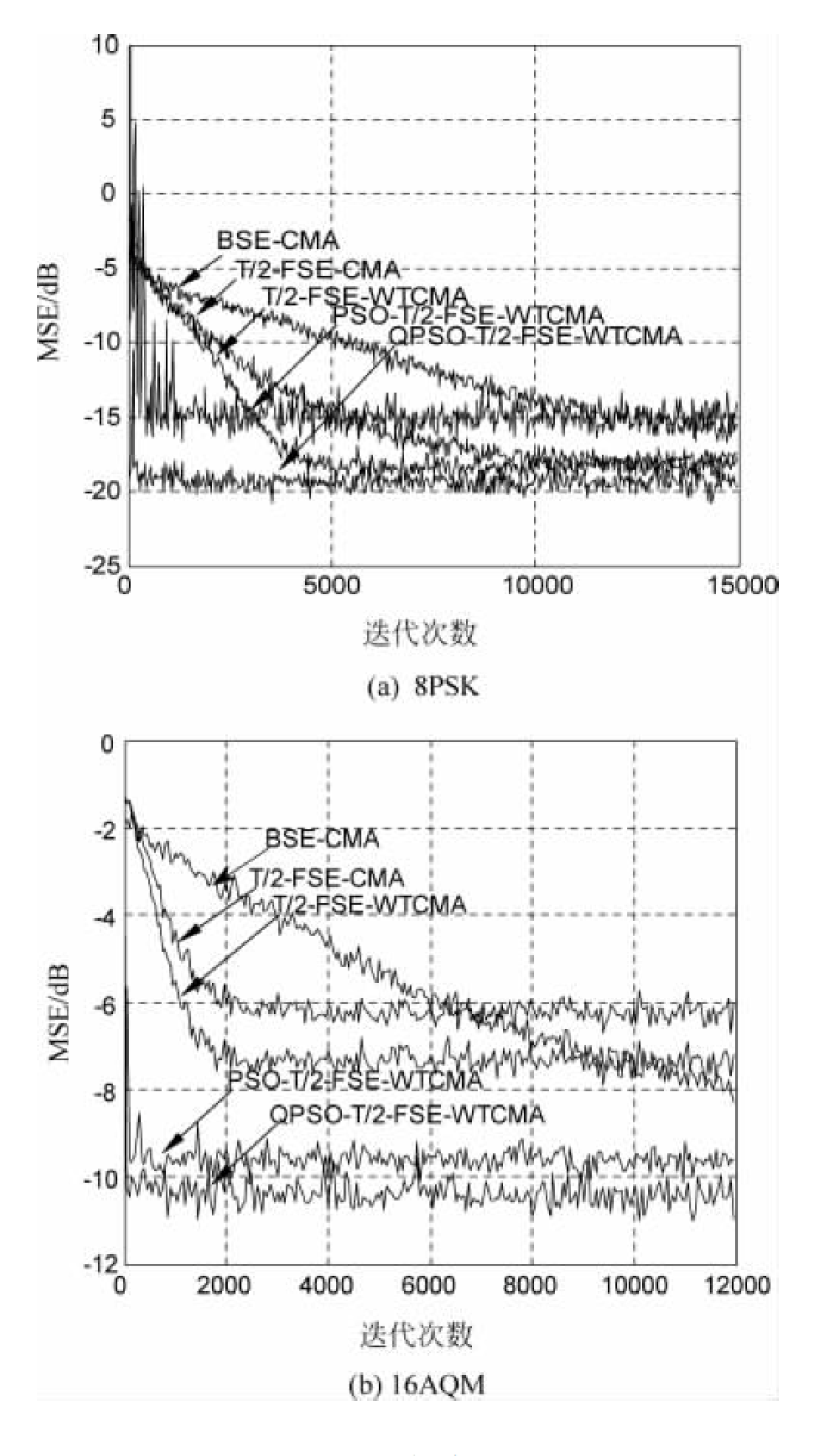



图 5.9　仿真结果

图 5.9(a)表明,在收敛速度上,QPSO-T/2-FSE-WTCMA 比 PSO-T/2-FSE-WTCMA 大约快了 1000 步,比 T/2-FSE-WTCMA 快了 4000 步;在稳态误差上,QPSO-T/2-FSE-WTCMA 与 PSO-T/2-FSE-WTCMA 减小了近 4dB,与 WT-T/2-FSE-CMA 和 T/2-FSE-CMA 相比,减小了近 1.5dB。图 5.9(b)表明,在收敛速度上,QPSO-T/2-FSE-WT-CMA 与 PSO-T/2-FSE-WTCMA 相当,与 T/2-FSE-WTCMA 和 T/2-FSE-CMA 相比快了近 1700 步;在稳态误差上,QPSO-T/2-FSE-WTCMA 比 PSO-T/2-FSE-WTCMA 减少了近 1dB,与 T/2-FSE-WTCMA 相比,减小了 3dB。

### 5.4.2 基于量子粒子群算法优化的正交小波多模盲均衡算法

由于多模盲均衡算法(MMA)均衡高阶 QAM 信号[26-27]存在收敛速度慢、稳态误差大的缺点,提出了基于量子粒子群算法优化的正交小波加权多模盲均衡算法(quantum particle swarm optimization algorithm based wavelet transform multi-modulus blind equalization algorithm,QPSO-WTWMMA)[28]。该算法根据高阶 QAM 信号星座图分布特点,选择合适的误差模型与高阶 QAM 星座图模型进行匹配,通过 QPSO 算法对加权多模盲均衡算法(weighted multi-modulus blind equalization algorithm,WMMA)的权向量进行优化,利用正交小波变换理论降低了输入信号的自相关性,从而提高了均衡高阶 QAM 信号的性能。

#### 1. WTMMA

在盲均衡算法中,CMA 能很好地均衡常数模信号,但很难均衡非常模的高阶 QAM 信号。这是因为用 CMA 均衡高阶 QAM 信号时,它将分布在几个半径不同圆上的输入星座图输出到同一个圆上,这实际上是将非常模高阶 QAM 信号处理为常模信号,导致 CMA 收敛后均方误差变大,严重影响了通信质量;同时 CMA 收敛速度慢,影响着均衡的实时性。为了有效均衡高阶 QAM 信号,文献[29-31]给出了多模盲均衡算法(MMA),该算

法将均衡器输出分成实部和虚部，在同相和正交方向上选取各自的模值，有效地克服了 CMA 单一判决圆造成调整误差大的缺点；而文献[32]给出的加权多模盲均衡算法（WMMA）通过充分利用星座图的先验信息，选择合适的误差模型匹配 QAM 星座图，进一步减小了稳态均方误差。为加快收敛速度，将正交小波变换引入 MMA 中，对输入信号进行变换，以降低输入信号的自相关性，提高收敛速度，这样得到的正交小波多模盲均衡算法原理（WTMMA），如图 5.10 所示。

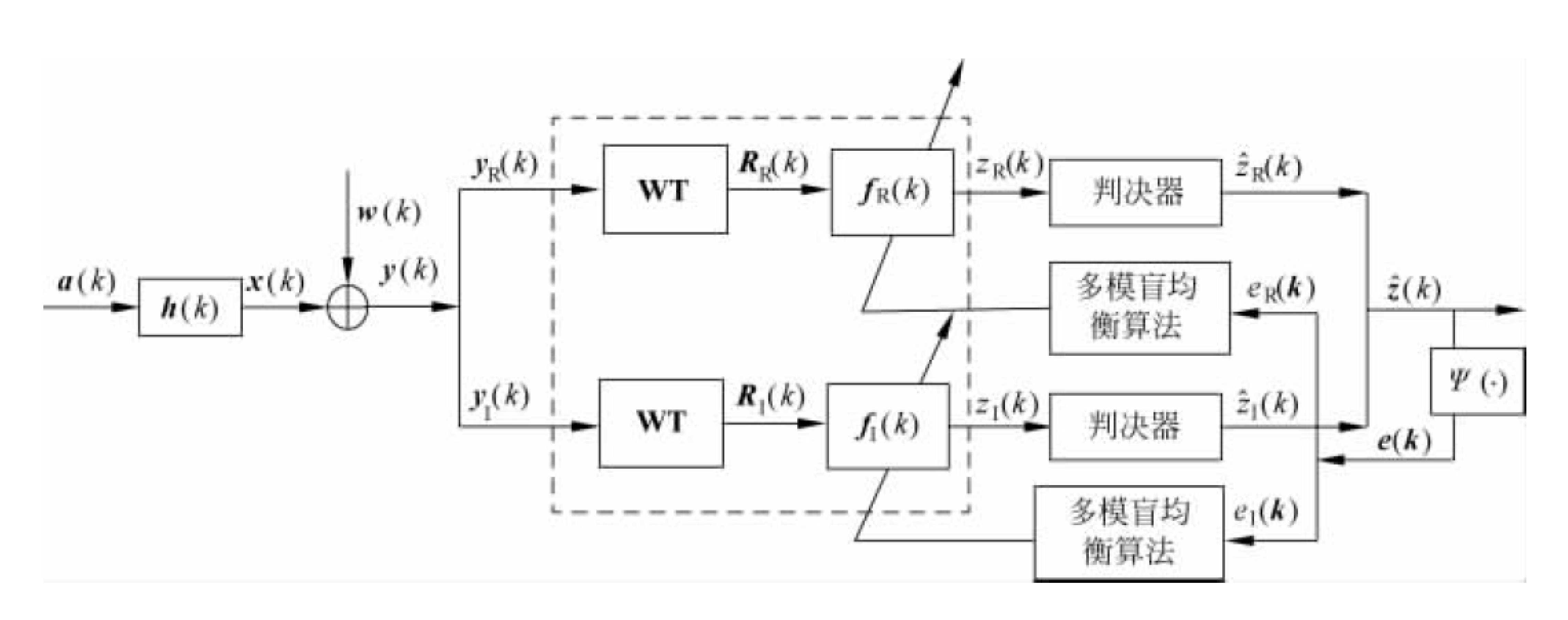



图 5.10　WTMMA 原理

图中，$\boldsymbol{a}(k)$是信源发射复信号；$\boldsymbol{h}(k)$是信道的脉冲响应向量；$\boldsymbol{w}(k)$是加性高斯白噪声向量；$\boldsymbol{y}(k)$是均衡器输入复信号向量；$\boldsymbol{R}(k)$是 $\boldsymbol{y}(k)$经过正交小波变换后的复信号；$\boldsymbol{f}(k)$是均衡器权向量；$z(k)$是均衡器的输出信号；$\hat{z}(k)$是判决器输出信号；$\boldsymbol{y}_{\mathrm{R}}(k)$和 $\boldsymbol{y}_{\mathrm{I}}(k)$分别为 $\boldsymbol{y}(k)$的实部和虚部；$\boldsymbol{\Psi}(\cdot)$是误差生成函数，$e(k)$是误差函数。

$$\boldsymbol{y}(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+\boldsymbol{w}(k) \tag{5.4.7a}$$

$$\boldsymbol{y}(k)=\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{y}_{\mathrm{I}}(k) \tag{5.4.7b}$$

经过小波变换后均衡器的输入信号 $\boldsymbol{R}(k)$表示为

$$\boldsymbol{R}(k)=\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{R}_{\mathrm{I}}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{V}\boldsymbol{y}_{\mathrm{I}}(k)) \tag{5.4.8}$$

式中，$\boldsymbol{R}_{\mathrm{R}}(k)$与 $\boldsymbol{R}_{\mathrm{I}}(k)$分别为输入信号 $\boldsymbol{R}(k)$的实部和虚部，即

$$\boldsymbol{R}_{\mathrm{R}}(k)=[u_{\mathrm{R}1,0}(k),u_{\mathrm{R}1,1}(k),\cdots,u_{\mathrm{R}J,k_J}(k),s_{\mathrm{R}J,0}(k),\cdots,s_{\mathrm{R}J,k_J}(k)]^{\mathrm{T}} \tag{5.4.9}$$

$$\begin{cases} u_{\mathrm{R}j,m}(k) = \sum_{l=0}^{L-1} y_{\mathrm{R}}(k-l)\varphi_{j,m}(l) \\ s_{\mathrm{R}J,m}(k) = \sum_{l=0}^{L-1} y_{\mathrm{R}}(k-l)\phi_{J,m}(l) \end{cases} \tag{5.4.10}$$

$$\boldsymbol{R}_{\mathrm{I}}(k) = [u_{\mathrm{I}1,0}(k), u_{\mathrm{I}1,1}(k), \cdots, u_{\mathrm{I}J,k_j}(k), s_{\mathrm{I}J,0}(k), \cdots, s_{\mathrm{I}J,k_j}(k)]^{\mathrm{T}} \tag{5.4.11}$$

$$\begin{cases} u_{\mathrm{I}j,m}(k) = \sum_{l=0}^{L-1} y_{\mathrm{I}}(k-l)\varphi_{j,m}(l) \\ s_{\mathrm{I}J,m}(k) = \sum_{l=0}^{L-1} y_{\mathrm{I}}(k-l)\phi_{J,m}(l) \end{cases} \tag{5.4.12}$$

式中，$u_{j,m}(k)$、$s_{J,m}(k)$分别表示相应的小波和尺度变换系数，$\boldsymbol{V}$ 为正交小波变换矩阵。

均衡器输出为

$$\begin{aligned} z(k) &= z_{\mathrm{R}}(k) + \mathrm{j}z_{\mathrm{I}}(k) \\ &= \boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{R}}(k) + \mathrm{j}(\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{I}}(k)) \end{aligned} \tag{5.4.13}$$

式中，$\boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}}(k)$和 $\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}}(k)$（上标 T 表示转置）分别为均衡器权向量 $\boldsymbol{f}^{\mathrm{T}}(k)$的实部和虚部；$z_{\mathrm{R}}(k)$和 $z_{\mathrm{I}}(k)$分别为均衡器输出信号 $z(k)$的实部和虚部。

MMA 的代价函数为

$$J_{\mathrm{MMA}} = E\{(z_{\mathrm{R}}^2(k) - R_{\mathrm{R,MMA}}^2)^2 + (z_{\mathrm{I}}^2(k) - R_{\mathrm{I,MMA}}^2)^2\} \tag{5.4.14}$$

式中，$R_{\mathrm{R,MMA}}^2 = E(a_{\mathrm{R}}^4(k))/E(a_{\mathrm{R}}^2(k))$，$R_{\mathrm{I,MMA}}^2 = E(a_{\mathrm{I}}^4(k))/E(a_{\mathrm{I}}^2(k))$，前者表示 $a(k)$实部模值，后者表示 $a(k)$虚部模值。

均衡器的误差函数为

$$\begin{cases} e_{\mathrm{R,MMA}}(k) = z_{\mathrm{R}}(k)(z_{\mathrm{R}}^2(k) - R_{\mathrm{R,MMA}}^2) \\ e_{\mathrm{I,MMA}}(k) = z_{\mathrm{I}}(k)(z_{\mathrm{I}}^2(k) - R_{\mathrm{I,MMA}}^2) \end{cases} \tag{5.4.15}$$

均衡器权向量的迭代公式为

$$\begin{cases} \boldsymbol{f}_{\mathrm{R}}(k+1) = \boldsymbol{f}_{\mathrm{R}}(k) - \mu\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1}(k)e_{\mathrm{R,MMA}}(k)\boldsymbol{R}_{\mathrm{R}}^{*}(k) \\ \boldsymbol{f}_{\mathrm{I}}(k+1) = \boldsymbol{f}_{\mathrm{I}}(k) - \mu\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1}(k)e_{\mathrm{I,MMA}}(k)\boldsymbol{R}_{\mathrm{I}}^{*}(k) \end{cases} \tag{5.4.16}$$

式中，$\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1}(k) = \mathrm{diag}[\sigma_{\mathrm{R}j,0}^2(k), \sigma_{\mathrm{R}j,1}^2(k), \cdots, \sigma_{\mathrm{R}J,k_j-1}^2(k), \sigma_{\mathrm{R}J+1,0}^2(k), \cdots,$

$\sigma^2_{\mathrm{R}J+1,k_j-1}(k)]$，$\sigma^2_{\mathrm{R}j,k}(k)$与$\sigma^2_{\mathrm{R}J+1,k_j}(k)$分别表示对小波变换系数实部$u_{\mathrm{R}j,m}(k)$、尺度变换系数实部$s_{\mathrm{R}J,m}(k)$的平均功率估计；$\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1}(k)=\mathrm{diag}[\sigma^2_{\mathrm{I}j,0}(k)$，$\sigma^2_{\mathrm{I}j,1}(k),\cdots,\sigma^2_{\mathrm{I}J,k_j-1}(k),\sigma^2_{\mathrm{I}J+1,0}(k),\cdots,\sigma^2_{\mathrm{I}J+1,k_j-1}(k)]$，$\sigma^2_{\mathrm{I}j,k}(k)$与$\sigma^2_{\mathrm{I}J+1,k_j}(k)$分别表示对小波变换系数虚部$u_{\mathrm{I}j,m}(k)$、尺度变换系数虚部$s_{\mathrm{I}J,m}(k)$的平均功率估计。其递推公式为

$$\begin{cases}\hat{\sigma}^2_{\mathrm{R}j,m}(k+1)=\beta\hat{\sigma}^2_{\mathrm{R}j,m}(k)+(1-\beta)\mid u_{\mathrm{R}j,m}(k)\mid^2\\ \hat{\sigma}^2_{\mathrm{R}J+1,m}(k+1)=\beta\hat{\sigma}^2_{\mathrm{R}J+1,m}(k)+(1-\beta)\mid s_{\mathrm{R}J,m}(k)\mid^2\end{cases}\tag{5.4.17}$$

$$\begin{cases}\hat{\sigma}^2_{\mathrm{I}j,m}(k+1)=\beta\hat{\sigma}^2_{\mathrm{I}j,m}(k)+(1-\beta)\mid u_{\mathrm{I}j,m}(k)\mid^2\\ \hat{\sigma}^2_{\mathrm{I}J+1,m}(k+1)=\beta\hat{\sigma}^2_{\mathrm{I}J+1,m}(k)+(1-\beta)\mid s_{\mathrm{I}J,m}(k)\mid^2\end{cases}\tag{5.4.18}$$

式中，$\beta$为平滑因子，且$0<\beta<1$。式(5.4.7)～式(5.4.18)就是正交小波多模盲均衡算法(WTMMA)。

2. QPSO-WTMMA

QPSO-WTMMA 的基本思想：随机初始化量子粒子的位置向量，并把位置向量作为 QPSO 算法的决策变量，将均衡器输入信号作为 QPSO 算法的输入信号，由 WTMMA 的代价函数确定 QPSO 算法的适应度函数，利用 QPSO 算法来寻找粒子的最优位置向量并作为 WTMMA 的初始化权向量。优化过程按 5.4.1 节所述进行，这里不再赘述。

3. 算法仿真

**【实验 5.5】** 为了检验 QPSO-WTMMA 的有效性，以 MMA 和 WTMMA 为比较对象，进行仿真实验。仿真实验中，种群规模为 40，最大进化代数为 100，中心抽头初始化。最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$，发射信号为 64QAM 信号；信噪比为 25dB；均衡器权长为 16。其他仿真参数如表 5.5 所示；100 次蒙特卡罗仿真结果如图 5.11 所示。

表 5.5 仿真参数

| 算　　法 | 步长 | 小波 | 级数 | 初始化功率 | $\beta$ 取值 |
|---|---|---|---|---|---|
| MMA | 0.000 005 | | | | |
| WTMMA | 0.0001 | DB4 | 3 | 4 | 0.999 |
| QPSO-WTMMA | 0.000 01 | DB4 | 3 | 4 | 0.999 |

(a) 均方误差

(b) MMA输出

(c) WTMMA输出

(d) QPSO-WTMMA输出

图 5.11 仿真结果

图 5.11(a)表明,在收敛速度上,QPSO-WTMMA 比 WTMMA 快了近 1000 步,与 MMA 相比,快了近 2000 步;在稳态误差上,QPSO-WTMMA

比 WTMMA 和 MMA 减少了近 2dB。图 5.11(b)、(c)、(d)表明，QPSO-WTMMA 的输出星座图更加清晰、紧凑。

### 5.4.3 基于量子粒子群算法优化的正交小波加权多模盲均衡算法

#### 1. WTWMMA

正交小波加权多模盲均衡算法(orthogonal wavelet transform weighted multi-modulus blind equalization algorithm，WTWMMA)是通过加权项自适应地修正模值，以进一步降低稳态误差，提高算法性能，而具体的加权因子是通过仿真实验确定的。

WTWMMA 的代价函数为

$$J_{\mathrm{WTWMMA}}=E\{(z_{\mathrm{R}}^2(k)-|\hat{z}_{\mathrm{R}}(k)|^{\lambda_{\mathrm{R}}}R_{\lambda_{\mathrm{R}},\mathrm{WTWMMA}}^2)^2+(z_{\mathrm{I}}^2(k)-|\hat{z}_{\mathrm{I}}(k)|^{\lambda_{\mathrm{I}}}R_{\lambda_{\mathrm{I}},\mathrm{WTWMMA}}^2)^2\} \tag{5.4.19}$$

式中，加权因子 $\lambda_{\mathrm{R}},\lambda_{\mathrm{I}}\in[0,2]$，对方形星座图，我们选择 $\lambda_{\mathrm{R}}=\lambda_{\mathrm{I}}=\lambda$，$\lambda_{\mathrm{R}}$ 和 $\lambda_{\mathrm{I}}$ 分别是信号实部和虚部的加权因子，仿真实验中，$\lambda$ 因子的值是根据仿真实例中经验得到的；$R_{\lambda_{\mathrm{R}},\mathrm{WTWMMA}}^2=E(a_{\mathrm{R}}^4(k))/E(|a_{\mathrm{R}}(k)|^{2+\lambda_{\mathrm{R}}})$，$R_{\lambda_{\mathrm{I}},\mathrm{WTWMMA}}^2=E(a_{\mathrm{I}}^4(k))/E(|a_{\mathrm{I}}(k)|^{2+\lambda_{\mathrm{I}}})$；判决信号 $\hat{z}_{\mathrm{R}}(k)$，$\hat{z}_{\mathrm{I}}(k)$ 是 $\hat{z}(k)$ 的实部与虚部。

均衡器误差函数为

$$\begin{cases}e_{\mathrm{R,WTWMMA}}(k)=z_{\mathrm{R}}(k)(z_{\mathrm{R}}^2(k)-|\hat{z}_{\mathrm{R}}(k)|^{\lambda_{\mathrm{R}}}R_{\lambda_{\mathrm{R}},\mathrm{WTWMMA}}^2)\\ e_{\mathrm{I,WTWMMA}}(k)=z_{\mathrm{I}}(k)(z_{\mathrm{I}}^2(k)-|\hat{z}_{\mathrm{I}}(k)|^{\lambda_{\mathrm{I}}}R_{\lambda_{\mathrm{I}},\mathrm{WTWMMA}}^2)\end{cases} \tag{5.4.20}$$

WTWMMA 均衡器权向量迭代公式为

$$\begin{cases}\boldsymbol{f}_{\mathrm{R}}(k+1)=\boldsymbol{f}_{\mathrm{R}}(k)-\mu\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1}(k)e_{\mathrm{R,WTWMMA}}(k)\boldsymbol{R}_{\mathrm{R}}^*(k)\\ \boldsymbol{f}_{\mathrm{I}}(k+1)=\boldsymbol{f}_{\mathrm{I}}(k)-\mu\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1}(k)e_{\mathrm{I,WTWMMA}}(k)\boldsymbol{R}_{\mathrm{I}}^*(k)\end{cases} \tag{5.4.21}$$

式(5.4.7)～式(5.4.13)，式(5.4.19)～式(5.4.21)构成了正交小波加权多模盲均衡算法(WTWMMA)。

## 2. QPSO-WTWMMA

(1) QPSO-WTWMMA 原理

将 QPSO 算法和正交小波变换运用于加权多模盲均衡算法(WMMA)中，通过 QPSO 算法对均衡器权向量进行优化，利用正交小波变换理论来降低输入信号的自相关性，这样得到的基于量子粒子群算法优化的正交小波加权常模盲均衡算法(QPSO based WTWMMA，QPSO-WTWMMA)原理，如图 5.12 所示。

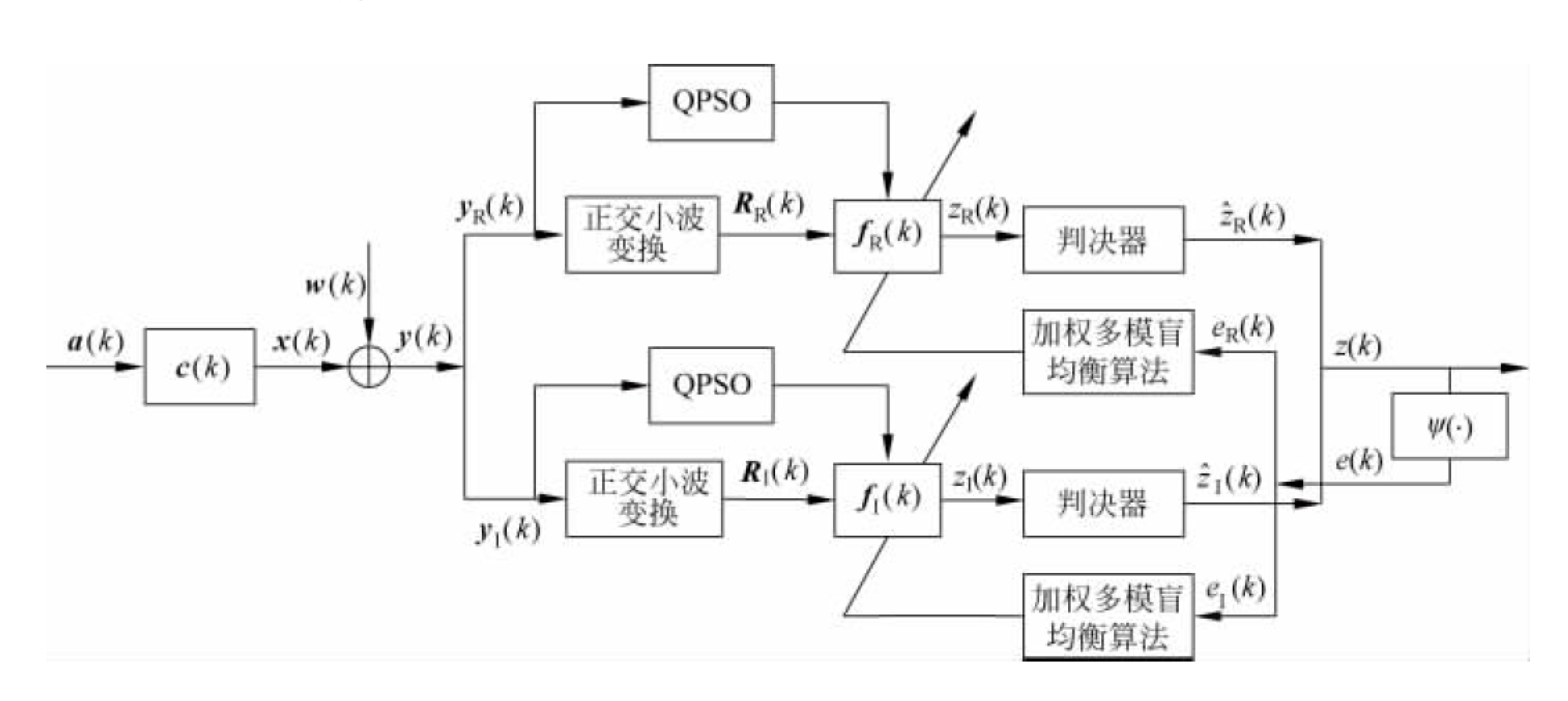



图 5.12 QPSO-WTWMMA 原理

(2) 确定适应度函数

QPSO 的适应度函数为

$$\text{Fit}(\boldsymbol{f}_i) = 1/J_{\text{WTWMMA}}(\boldsymbol{f}_i), \quad i = 1,2,\cdots,N \tag{5.4.22}$$

式中，$J_{\text{WTWMMA}}(\boldsymbol{f}_i)$是加权多模盲均衡算法的代价函数，$\boldsymbol{f}_i$ 是粒子的位置向量。

(3) 优化过程

优化过程按 5.4.1 节所述进行，这里不再赘述。

## 3. 算法仿真

为了检验 QPSO-WTWMMA 的有效性，以 QPSO-WTMMA、WTWMMA、

WTMMA 和 MMA 为比较对象。实验中，种群规模为 40，最大进化代数为 500，由于 QPSO 算法对权向量初始化时具有随机性，因此对 20 次独立实验仿真结果放在一起进行对比。

**【实验 5.6】** 混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$，发射信号为 256QAM，均衡器权长为 16，信噪比为 30dB；信道输入信号采用 DB2 正交小波进行分解，分解层次是 2，功率初始值设置为 4，遗忘因子 $\beta=0.99$；加权因子 $\lambda=1.2$，步长 $\mu_{\text{MMA}}=6\times10^{-7}$，$\mu_{\text{WTMMA}}=8\times10^{-5}$，$\mu_{\text{WTWMMA}}=3\times10^{-4}$，$\mu_{\text{QPSO-WTMMA}}=3\times10^{-5}$，$\mu_{\text{QPSO-WTWMMA}}=5\times10^{-5}$，对 20 次独立实验仿真结果放在一起进行对比，如图 5.13 和图 5.14 所示。

图 5.13(a)～(e)表明，在每次仿真过程中，由于高斯噪声的随机性，使得均方误差曲线具有抖动，但其收敛趋势是一致的，这说明算法的稳定性与适应性较强；由图 5.13(d)、(e)所示的均方误差离散值比图 5.13(a)、(b)、(c)的离散值要大一些，这可能是由于 QPSO 算法初始化粒子的位置向量和 QPSO 算法输入信号的决策变量(即为均衡器输入信号向量)均是随机向量所致。但从图 5.13(f)可知，利用 QPSO 算法来寻找最优权向量后，QPSO -WTWMMA 的收敛速度比 QPSO -WTWMMA、WTWMMA、WTMMA 及 MMA 分别快了近 0 步、4000 步、5000 步、7000 步；而稳态误差分别减小了 1dB、3dB、3dB、3dB。可见，虽然 QPSO 算法初始化粒子位置向量的随机性可能导致 QPSO-WTWMMA 与 QPSO-WTMMA 均方误差离散值较大，但与其他算法相比，QPSO-WTWMMA 的稳态均方误差最小、收敛速度最快，均衡性能最优。

图 5.14 表明，QPSO -WTWMMA 的输出星座图最清晰、紧凑，而 QPSO-WTMMA 的输出星座比 WTWMMA 清晰，WTWMMA 的星座图又比 WTMMA 清晰紧凑。因此，引入 QPSO 算法、正交小波变换与 WMMA 后，算法性能得到较大提高，可以正确地恢复传输信号，具有一定的实用性。

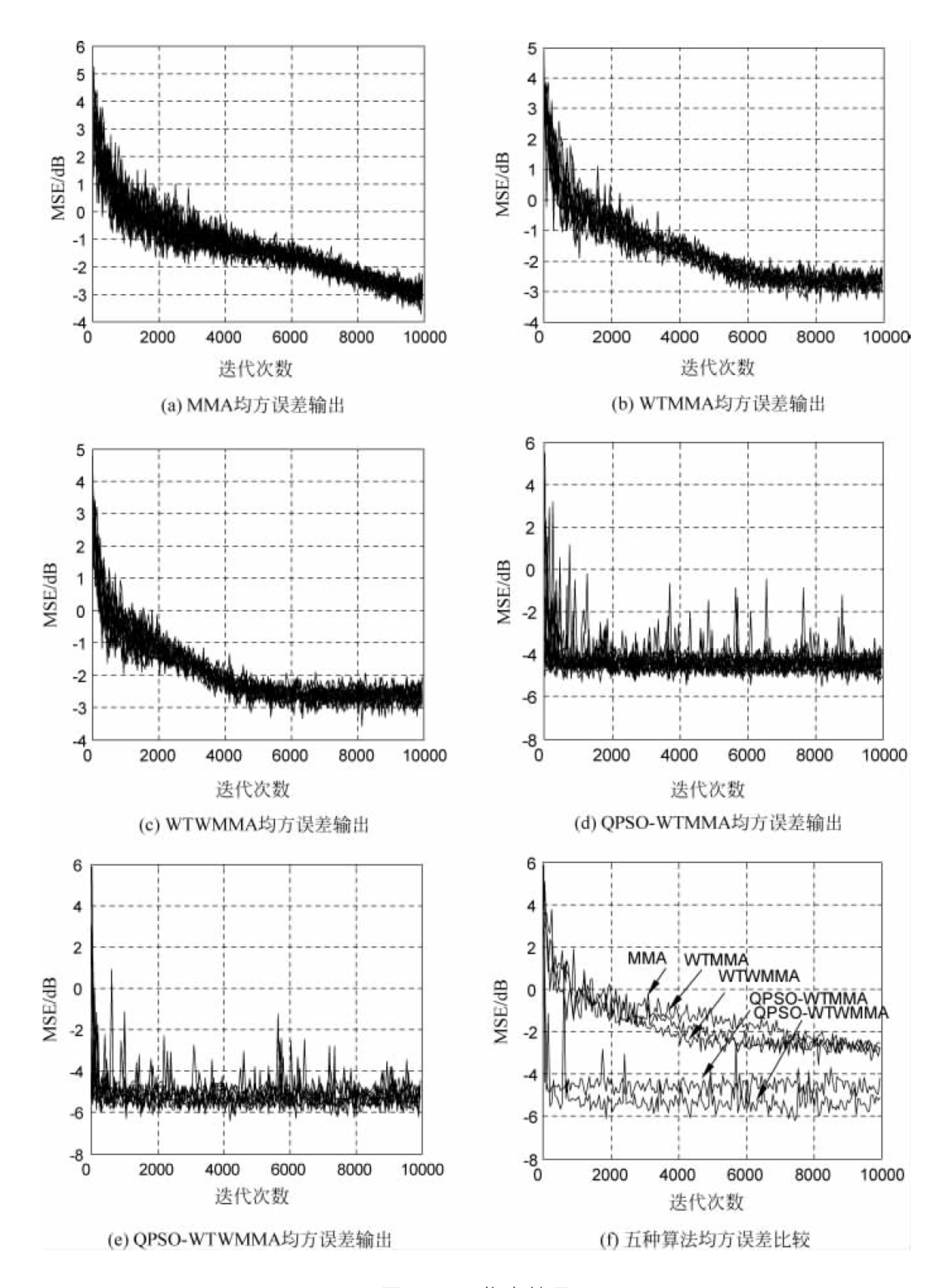

MSE/dB
迭代次数
(a) MMA均方误差输出
(b) WTMMA均方误差输出
(c) WTWMMA均方误差输出
(d) QPSO-WTMMA均方误差输出
(e) QPSO-WTWMMA均方误差输出
MMA
WTMMA
WTWMMA
QPSO-WTMMA
QPSO-WTWMMA
(f) 五种算法均方误差比较

图 5.13 仿真结果

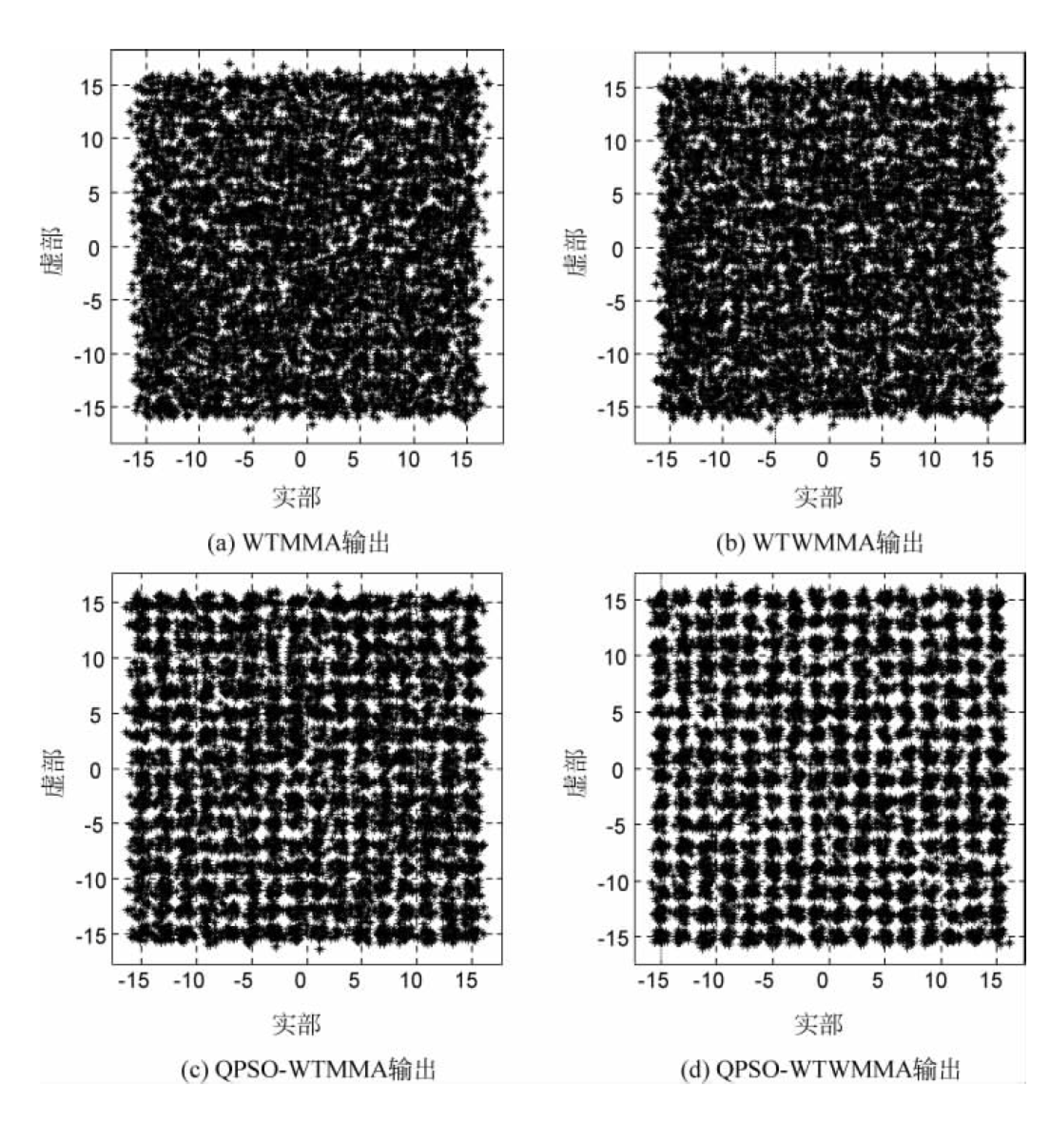



图 5.14 仿真结果

## 5.5 基于动态粒子群算法优化的正交小波加权多模盲均衡算法

动态粒子群优化(dynamic particle swarm optimization, DPSO)算法[33-35]是一种随机全局搜索的优化算法,是标准 PSO 算法的又一改进算法,该算法探测到外部环境变化时,将按种群中的一部分比例来重新初始化粒子的位置向量和速度向量,具有良好的追踪动态极值能力,保持了进化后期粒子种群的多样性。

### 5.5.1 基于动态粒子群算法优化的正交小波多模盲均衡算法

#### 1. DPSO-WTWMMA 描述

(1) 基本思想

将动态粒子群优化算法引入到正交小波多模盲均衡算法中,得到基于动态粒子群算法优化的正交小波加权多模盲均衡算法(dynamic particle swarm optimization based WTWMMA,DPSO-WTWMMA)[36]。其基本思想是将随机粒子种群的位置向量(均衡器初始化权向量)作为 DPSO 算法的决策变量;将均衡器输入信号作为 DPSO 算法的输入信号,由 WTWMMA 的代价函数,确定 DPSO 算法的适应度函数;通过 DPSO 算法寻找粒子群的最优位置向量,并作为正交小波多模盲均衡算法(WTWMMA)的初始化权向量。将 WTWMMA 代价函数的倒数作为 DPSO 算法的适应度函数,则

$$\mathrm{Fit}(\boldsymbol{f}_i) = 1/J_{\mathrm{WTWMMA}}(\boldsymbol{f}_i),\quad i = 1,2,\cdots,M \tag{5.5.1}$$

式中,$J_{\mathrm{WTWMMA}}(f_i) = E\{(z_{\mathrm{R}}^2(k) - R_{\mathrm{R,WTWMMA}}^2)^2 + (z_{\mathrm{I}}^2(k) - R_{\mathrm{I,WTWMMA}}^2)^2\}$ 是 WTWMMA 的代价函数,$R_{\mathrm{R,WTWMMA}}^2 = E(a_{\mathrm{R}}^4(k))/E(|a_{\mathrm{R}}(k)|^2)$,$R_{\mathrm{I,WTWMMA}}^2 = E(a_{\mathrm{I}}^4(k))/E(|a_{\mathrm{I}}(k)|^2)$;$\boldsymbol{f}_i$ 是粒子的位置向量,对应于均衡器权向量个体。

(2) 动态粒子群算法优化权向量方法

同 PSO 算法一样,DPSO 算法也采用位置和速度搜索模型,每个粒子对应于所求问题的一个候选解,适应度值决定解的优劣。初始化第 $i$ 个粒子的位置向量 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ 和速度 $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$,$x_{id}$ 和 $v_{id}$ 分别表示第 $i$ 个粒子第 $d$ 维位置和第 $i$ 个粒子第 $d$ 维速度。设随机产生的初始粒子群为 $\boldsymbol{f} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_m]$,其中的每一个粒子 $\boldsymbol{f}_i$ $(0 < i \leqslant m)$ 对应于均衡器的一个权向量。粒子更新自己的速度和位置的公式为[10]

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1(p_{id}(t) - x_{id}(t)) + c_2 r_2(p_{gd}(t) - x_{id}(t)) \tag{5.5.2}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{5.5.3}$$

$$w = w_{\max} - ((w_{\max} - w_{\min})/M)t \tag{5.5.4}$$

式中，$i=1,2,\cdots,N$，$d=1,2,\cdots,D$，$D$ 为空间维数；$c_1$ 和 $c_2$ 为加速因子；$r_1$ 和 $r_2$ 为在[0,1]内变化的随机数；$M$ 为最大迭代次数；$w$ 为惯性权重，$w_{max}$ 和 $w_{min}$ 分别为最大的和最小的惯性权重。粒子 $i$ 在寻优过程中通过动态跟踪两个极值来更新位置和速度向量，当前迭代次数产生最优值即个体最优极值 $\boldsymbol{p}_i=(p_{i1},p_{i2},\cdots,p_{iD})$，$p_{id}$ 表示第 $i$ 个粒子第 $d$ 维个体极值，整个粒子群当前的全局最优极值 $\boldsymbol{p}_g=(p_{g1},p_{g2},\cdots,p_{gD})$，$p_{gd}$ 表示第 $i$ 个粒子第 $d$ 维全局极值。

因此，对标准 PSO 算法进行两个方面的改进：

(1) 引入探测机制，使粒子获得感知外界环境变化的能力。在 $D$ 维空间中设置 $n_1$ 个粒子和 $n_2$ 个敏感粒子($n_1\neq n_2$)，每次迭代时都计算粒子的适应度值 Fit 和敏感粒子的适应度值 Fitness，并求出敏感粒子适应度值累加和 $o$Fitness。通过比较个体最优和全局最优位置向量，计算相邻两次敏感粒子适应度值累加和之差值即 $\Delta F=o\text{Fitness}-n\text{Fitness}$，其中 $n$Fitness 为敏感粒子第二次初始化后的适应度值累加和。当 $\Delta F$ 不为 0 时，外界环境发生变化。

(2) 引入响应机制，当探测环境变化后，采用某种响应机制对种群进行更新即按一定比例重新初始化粒子的位置和速度向量，以适应动态环境。

以上描述用程序表示如下：如果满足条件：abs($o$Fitness－$n$Fitness)＞0，则

```
index = randperm(n1);
x(index(1:10),:) = rand(10,D);
v(index(1:10),:) = rand(10,D);
```

通过以上可知，当 abs($o$Fitness－$n$Fitness)＞0 时，就对粒子的位置向量和速度向量按一定比例重新进行初始化，使种群在新的粒子群体中重新寻找最优位置向量，以保持种群多样性，避免种群进化后期容易陷入“早熟”收敛。在 DPSO 算法中，将迭代最后时寻找粒子的最优位置向量作为均衡器最优权向量，即 $\boldsymbol{p}_{gbest}=(p_{g1},p_{g2},\cdots,p_{gD})$ 作为 WTWMMA 的初始化权向量。DPSO-WTWMMA 原理如图 5.15 所示。

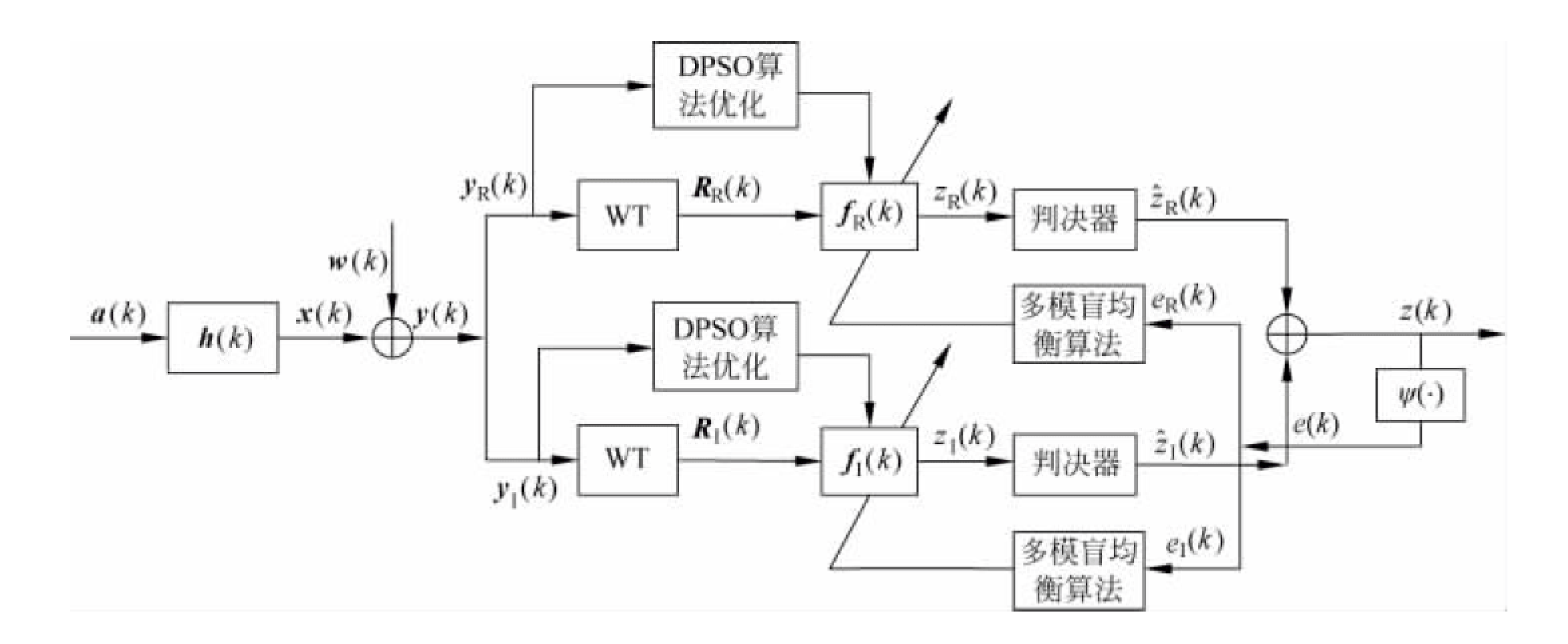



图 5.15 DPSO-WTMMA 原理

图中各量关系如式(5.4.7)～式(5.4.18)所示。

2. 算法仿真

为了检验 DPSO-WTMMA 的有效性,以 DPSO-MMA 和 MMA 为比较对象,进行仿真实验。

**【实验 5.7】** 混合相位水声信道 $\boldsymbol{h}=[0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$,发射信号为 64QAM,均衡器权长为 16,信噪比为 25dB;信道输入信号采用 DB2 正交小波进行分解,分解层次是 2,功率初始值设置为 4,遗忘因子 $\beta=0.999$;MMA 中第 6 个抽头系数设置为 1,其余为 0;其步长 $\mu_{\text{MMA}}=6\times10^{-6}$,$\mu_{\text{DPSO-MMA}}=2\times10^{-6}$,$\mu_{\text{DPSO-WTMMA}}=1.5\times10^{-5}$;100 次蒙特卡罗仿真结果如图 5.16 所示。

图 5.16 表明,DPSO-WTMMA 的收敛速度最快,比 MMA 快约 1500 步;而 DPSO-WTMMA 的稳态均方误差最小约为 −7dB,DPSO-MMA 收敛后均方误差为 −5dB,MMA 为 −3dB;DPSO-WTMMA 的输出星座图比 MMA、DPSO-MMA 更为清晰、紧凑。这表明 DPSO-WTMMA 有最快的收敛速度、最清晰的输出星座图和最小的稳态均方误差。

**【实验 5.8】** 两径水声信道 $\boldsymbol{h}=[-0.35 \quad 0 \quad 0 \quad 1]$,发射信号为 128QAM,均衡器权长为 16,信噪比为 30dB;信道输入信号采用 DB2 正交

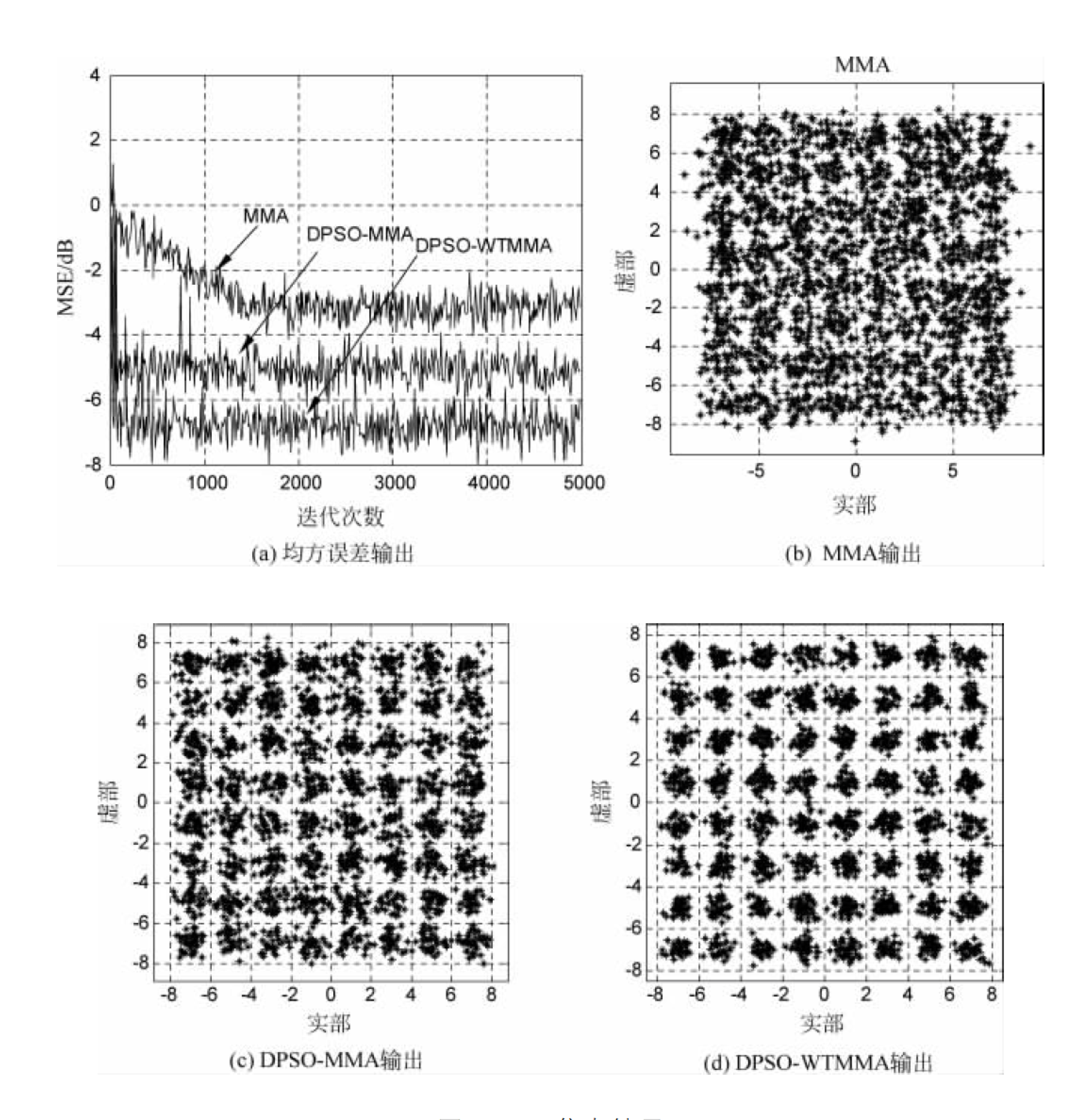



图 5.16　仿真结果

小波进行分解，分解层次是 3，功率初始值设置为 8，遗忘因子 $\beta=0.999$；其步长 $\mu_{\mathrm{MMA}}=8\times10^{-7}$，$\mu_{\mathrm{DPSO\text{-}MMA}}=2\times10^{-7}$，$\mu_{\mathrm{DPSO\text{-}WTMMA}}=3.5\times10^{-6}$。100 次蒙特卡罗仿真结果如图 5.17 所示。

图 5.17 表明，DPSO-WTMMA 的收敛速度最快，比 MMA 快约 1000 步；而 DPSO-WTMMA 的稳态均方误差最小约为－8dB，DPSO-MMA 收敛后的均方误差为－6dB，MMA 为－2dB；DPSO-WTMMA 的输出星座图比 MMA、DPSO-MMA 更为清晰、紧凑。这表明，DPSO-WTMMA 有最快的收敛速度、最清晰的输出星座图和最小的稳态均方误差。

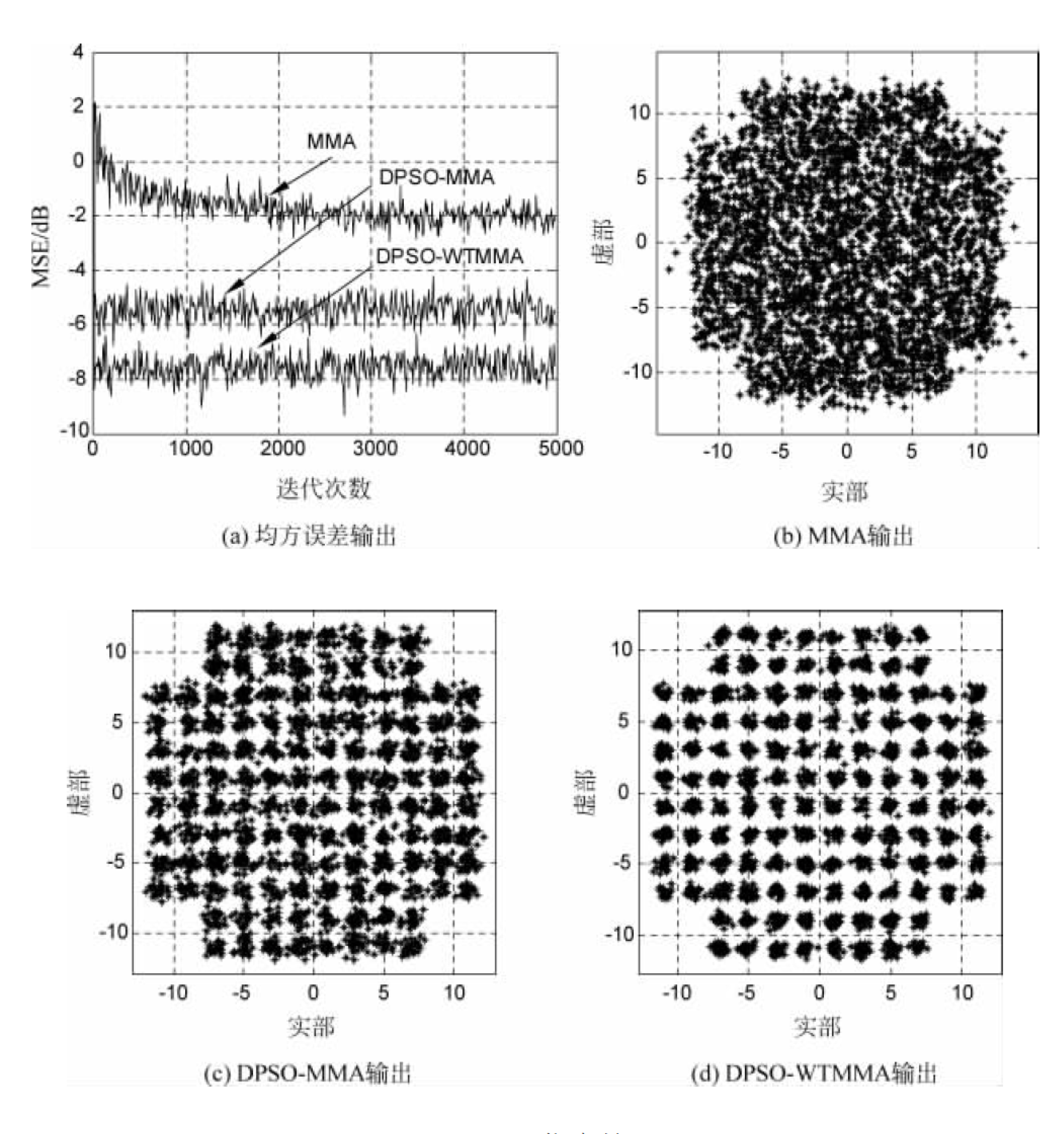



图 5.17 仿真结果

### 5.5.2 基于动态粒子群算法优化的正交小波动态加权多模盲均衡算法

#### 1. WTDWMMA 描述

动态加权多模算法（dynamic weighted multi-modulus blind equalization algorithm，DWMMA）的思想是通过充分利用星座图的先验信息来选择合适的误差模型匹配 QAM 星座图，以进一步减小稳态均方误差。将正交小波变换理论引入到 DWMMA，得到正交小波动态加权多模盲均衡

算法（orthogonal wavelet transform based dynamic weighted multi-modulus blind equalization algorithm，WTDWMMA）。其原理图如图 5.18 所示。

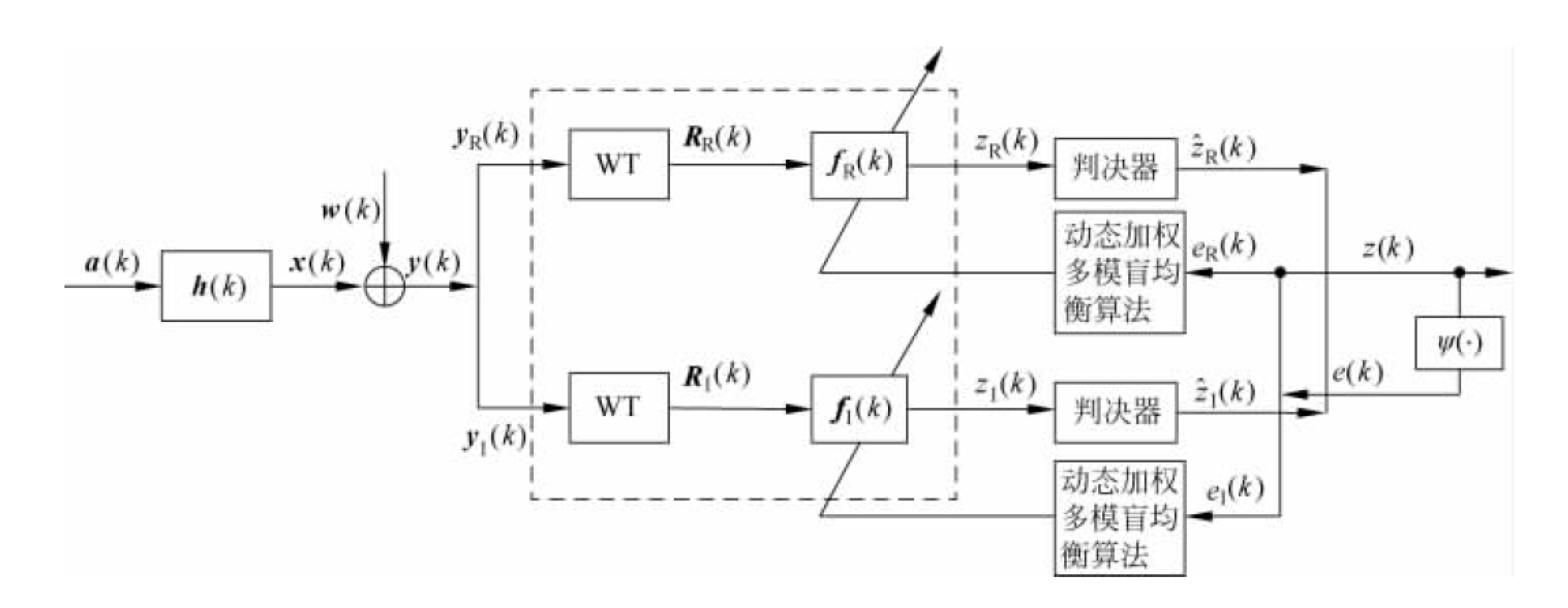



图 5.18　WTDWMMA 原理

由图 5.18 得，均衡器输入信号为

$$\boldsymbol{y}(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+\boldsymbol{w}(k) \tag{5.5.5}$$

均衡器权向量为

$$\boldsymbol{f}(k)=\boldsymbol{f}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{f}_{\mathrm{I}}(k) \tag{5.5.6}$$

经过小波变换后均衡器的输入信号表示为

$$\boldsymbol{R}(k)=\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{R}_{\mathrm{I}}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{V}\boldsymbol{y}_{\mathrm{I}}(k)) \tag{5.5.7}$$

式中，$\boldsymbol{R}_{\mathrm{R}}(k)$与$\boldsymbol{R}_{\mathrm{I}}(k)$分别为输入信号$\boldsymbol{R}(k)$的实部和虚部。

均衡器输出为

$$\begin{aligned}z(k)&=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k)\\&=\boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{I}}(k))\end{aligned} \tag{5.5.8}$$

式中，$\boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}}(k)$和$\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}}(k)$（上标 T 表示转置）分别为均衡器权向量的实部和虚部；$z_{\mathrm{R}}(k)$和$z_{\mathrm{I}}(k)$分别为均衡器输出信号的实部和虚部。

WTDWMMA 的代价函数为

$$\begin{aligned}J_{\mathrm{WTDWMMA}}=E\{&(z_{\mathrm{R}}^{2}(k)-|\hat{z}_{\mathrm{R}}(k)|^{\lambda(k)}R_{\mathrm{R,WTDWMMA}}^{2})^{2}+\\&(z_{\mathrm{I}}^{2}(k)-|\hat{z}_{\mathrm{I}}(k)|^{\lambda(k)}R_{\mathrm{I,WTDWMMA}}^{2})^{2}\}\end{aligned} \tag{5.5.9}$$

式中，$R_{\mathrm{R,WTDWMMA}}^{2}=E(a_{\mathrm{R}}^{4}(k))/E(|a_{\mathrm{R}}(k)|^{2+\lambda(k)})$，$R_{\mathrm{I,WTDWMMA}}^{2}=E(a_{\mathrm{I}}^{4}(k))/$

$E(|a_I(k)|^{2+\lambda(k)})$；判决信号$\hat{z}_R(k)$，$\hat{z}_I(k)$是$\hat{z}(k)$的实部与虚部；$\lambda(k)$为加权因子，决定算法的稳态误差和收敛速度。WTDWMMA 通过迭代过程中稳态误差 MSE 来确定 $\lambda(k)$，MSE 的迭代公式为

$$\text{MSE}(k+1) = \alpha\text{MSE}(k) + (1-\alpha)\,|\,z(k) - \hat{z}(k)\,|^2 \tag{5.5.10}$$

式中，$\alpha$ 为遗传因子，$\alpha=0.99$。

均衡器的误差函数为

$$\begin{cases} e_{\text{R,WTDWMMA}}(k) = z_R(k)(z_R^2(k) - |\,\hat{z}_R(k)\,|^{\lambda(k)} R_{\text{R,WTDWMMA}}^2) \\ e_{\text{I,WTDWMMA}}(k) = z_I(k)(z_I^2(k) - |\,\hat{z}_I(k)\,|^{\lambda(k)} R_{\text{I,WTDWMMA}}^2) \end{cases} \tag{5.5.11}$$

由式(5.5.11)知，$\lambda(k)$的取值直接影响了误差的大小，在均衡的初始阶段，信号眼图是完全闭合的，$\lambda(k)$应取较小的值，使误差较大以达到全局收敛；随着眼图的张开，需要不断增大 $\lambda(k)$的值来降低误差，利用更加精确的误差模型去匹配实际信号，达到降低稳态误差的目的。$\lambda(k)$定义为

$$\lambda(k) = \begin{cases} 0, & \text{MSE}(k) > d/2 \\ 1-\text{MSE}(k), & d/2 \geqslant \text{MSE}(k) > 2d/5 \\ 1.2-\text{MSE}(k), & 2d/5 \geqslant \text{MSE}(k) > 3d/10 \\ 1.5-\text{MSE}(k), & 3d/10 \geqslant \text{MSE}(k) > d/5 \\ 1.8-\text{MSE}(k), & d/5 \geqslant \text{MSE}(k) > d/10 \\ 2, & d/10 \geqslant \text{MSE}(k) \end{cases} \tag{5.5.12}$$

式中，$d$ 为星座点间的最小距离即最小欧几里得距离，该参数反映了 QAM 信号抗高斯白噪声的能力，由文献[36]知，未经过能量归一化的 QAM 星座图中，星座点能量的均值和峰值分别为

$$E = \sum_{i=1}^{M}(a_i + b_i)/M_C \tag{5.5.13}$$

$$P = \max(a_i + b_i) \tag{5.5.14}$$

式中，$a_i$，$b_i$ 表示 QAM 信号星座点的位置；$M_C$ 为星座图中任意象限星座点的个数。峰均比值为 $\gamma=P/E$，对星座图进行能量归一化后，相邻星座点之间的最小距离为 $d=2/\sqrt{E}$。

均衡器权向量的迭代公式为

$$\begin{cases} \boldsymbol{f}_{\mathrm{R}}(k+1) = \boldsymbol{f}_{\mathrm{R}}(k) - \mu \hat{\boldsymbol{R}}_{\mathrm{R}}^{-1}(k) e_{\mathrm{R,WTDWMMA}}(k) \boldsymbol{R}_{\mathrm{R}}^{*}(k) \\ \boldsymbol{f}_{\mathrm{I}}(k+1) = \boldsymbol{f}_{\mathrm{I}}(k) - \mu \hat{\boldsymbol{R}}_{\mathrm{I}}^{-1}(k) e_{\mathrm{I,WTDWMMA}}(k) \boldsymbol{R}_{\mathrm{I}}^{*}(k) \end{cases} \tag{5.5.15}$$

式中，$\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1}(k) = \mathrm{diag}[\sigma_{\mathrm{R}j,0}^{2}(k), \sigma_{\mathrm{R}j,1}^{2}(k), \cdots, \sigma_{\mathrm{R}J,k_j-1}^{2}(k), \sigma_{\mathrm{R}J+1,0}^{2}(k), \cdots, \sigma_{\mathrm{R}J+1,k_j-1}^{2}(k)]$，$\sigma_{\mathrm{R}j,k}^{2}(k)$与$\sigma_{\mathrm{R}J+1,k_j}^{2}(k)$分别表示对小波变换系数实部$u_{\mathrm{R}j,m}(k)$、尺度变换系数实部$s_{\mathrm{R}J,m}(k)$的平均功率估计；$\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1}(k) = \mathrm{diag}[\sigma_{\mathrm{I}j,0}^{2}(k), \sigma_{\mathrm{I}j,1}^{2}(k), \cdots, \sigma_{\mathrm{I}J,k_j-1}^{2}(k), \sigma_{\mathrm{I}J+1,0}^{2}(k), \cdots, \sigma_{\mathrm{I}J+1,k_j-1}^{2}(k)]$，$\sigma_{\mathrm{I}j,k}^{2}(k)$与$\sigma_{\mathrm{I}J+1,k_j}^{2}(k)$分别表示对小波变换系数虚部$u_{\mathrm{I}j,m}(k)$、尺度变换系数虚部$s_{\mathrm{I}J,m}(k)$的平均功率估计。其递推公式为

$$\begin{cases} \hat{\sigma}_{\mathrm{R}j,m}^{2}(k+1) = \beta \hat{\sigma}_{\mathrm{R}j,m}^{2}(k) + (1-\beta) \left| u_{\mathrm{R}j,m}(k) \right|^{2} \\ \hat{\sigma}_{\mathrm{R}J+1,m}^{2}(k+1) = \beta \hat{\sigma}_{\mathrm{R}J+1,m}^{2}(k) + (1-\beta) \left| s_{\mathrm{R}J,m}(k) \right|^{2} \end{cases} \tag{5.5.16}$$

$$\begin{cases} \hat{\sigma}_{\mathrm{I}j,m}^{2}(k+1) = \beta \hat{\sigma}_{\mathrm{I}j,m}^{2}(k) + (1-\beta) \left| u_{\mathrm{I}j,m}(k) \right|^{2} \\ \hat{\sigma}_{\mathrm{I}J+1,m}^{2}(k+1) = \beta \hat{\sigma}_{\mathrm{I}J+1,m}^{2}(k) + (1-\beta) \left| s_{\mathrm{I}J,m}(k) \right|^{2} \end{cases} \tag{5.5.17}$$

式中，$\beta$为平滑因子，且$0<\beta<1$。

### 2. DPSO-WTDWMMA 原理

DPSO-WTDWMMA 的原理如图 5.19 所示。

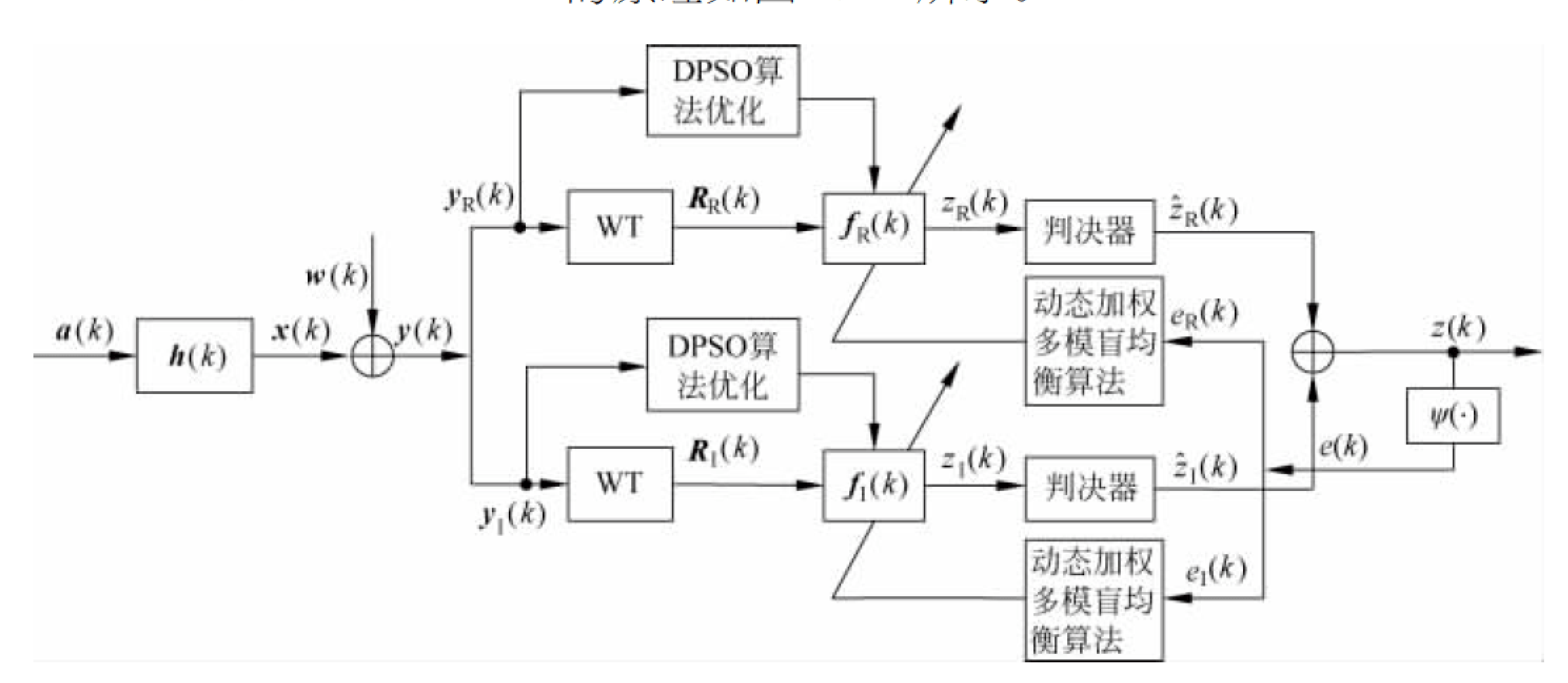



图 5.19 DPSO-WTDWMMA 原理

3. 算法仿真

为了检验 DPSO -WTDWMMA 的有效性,以 DPSO -DWMMA 和 DWMMA 为比较对象,进行仿真实验。

**【实验 5.9】** 混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$,发射信号为 64QAM,均衡器权长为 16,信噪比为 25dB;信道输入信号采用 DB2 正交小波进行分解,分解层次是 2,功率初始值设置为 10,遗忘因子 $\beta=0.999$;MSE(1)=2,DWMMA 中第 8 个抽头系数设置为 1,其余为 0;其步长 $\mu_{\text{DWMMA}}=3\times10^{-6}$,$\mu_{\text{DPSO-DWMMA}}=2.5\times10^{-6}$,$\mu_{\text{DPSO-WTDWMMA}}=1\times10^{-5}$;100 次蒙特卡罗仿真结果如图 5.20 所示。

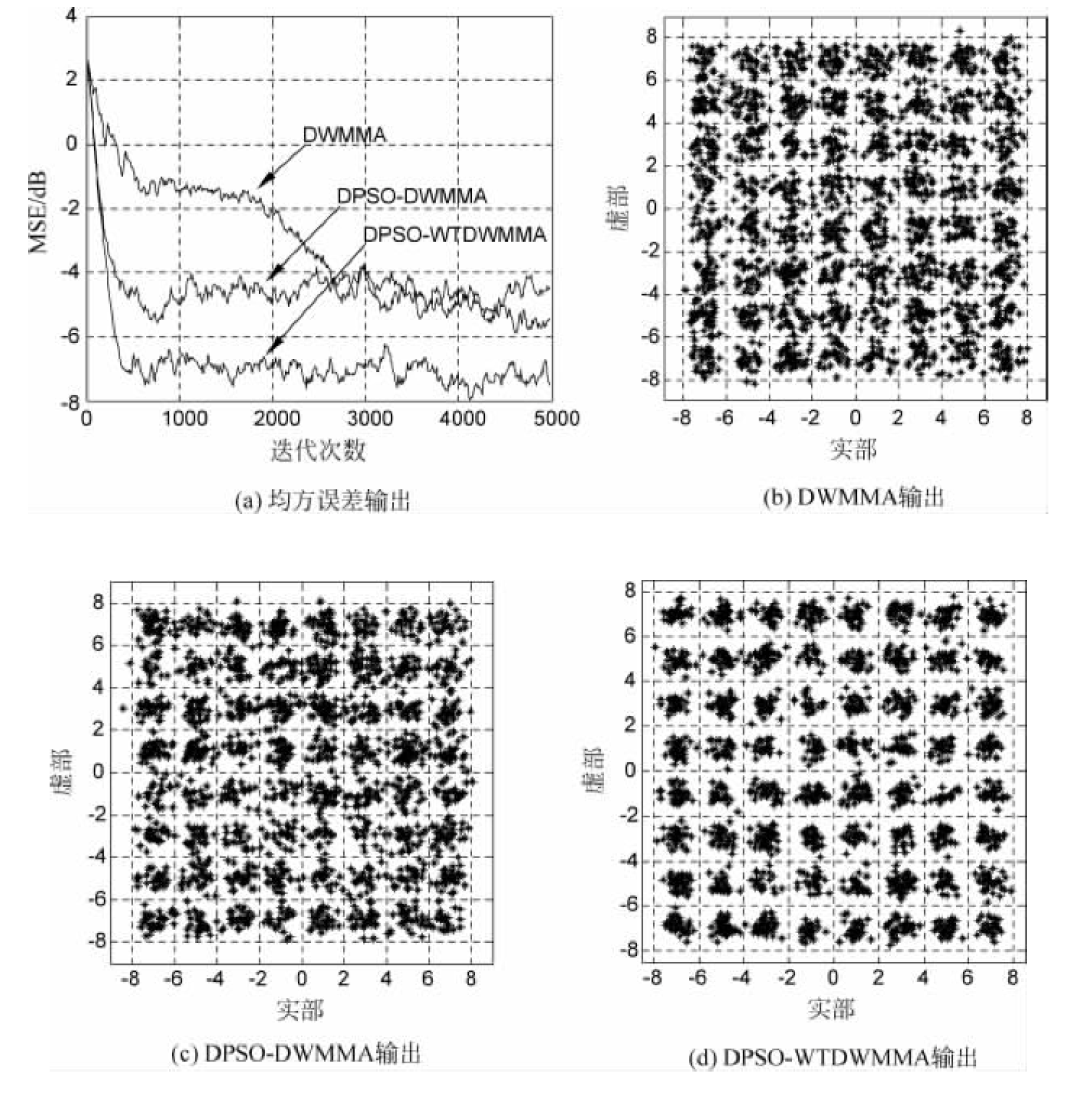



图 5.20 仿真结果

图 5.20 表明，DPSO-WTDWMMA 的收敛速度最快，比 WMMA 快约 2000 步、比 DPSO-WMMA 快约 500 步；而 DPSO-WWMMA 的稳态均方误差最小约为 −7dB，DPSO-WMMA 收敛后的均方误差为 −4.5dB，WMMA 为 −4dB；DPSO-WTDWMMA 的输出星座图比 WMMA、DPSO-WMMA 更为清晰、紧凑。

**【实验 5.10】** 最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.0578\quad 0.2368]$，发射信号为 128QAM，均衡器权长为 16，信噪比为 30dB；信道输入信号采用 DB2 正交小波进行分解，分解层次是 3，功率初始值设置为 10，遗忘因子 $\beta=0.99$，MSE(1)＝2；其步长 $\mu_{\text{DWMMA}}=5\times10^{-7}$，$\mu_{\text{DPSO-DWMMA}}=2\times10^{-7}$，$\mu_{\text{DPSO-WTDWMMA}}=3\times10^{-6}$；20 次蒙特卡罗仿真结果如图 5.21 所示。

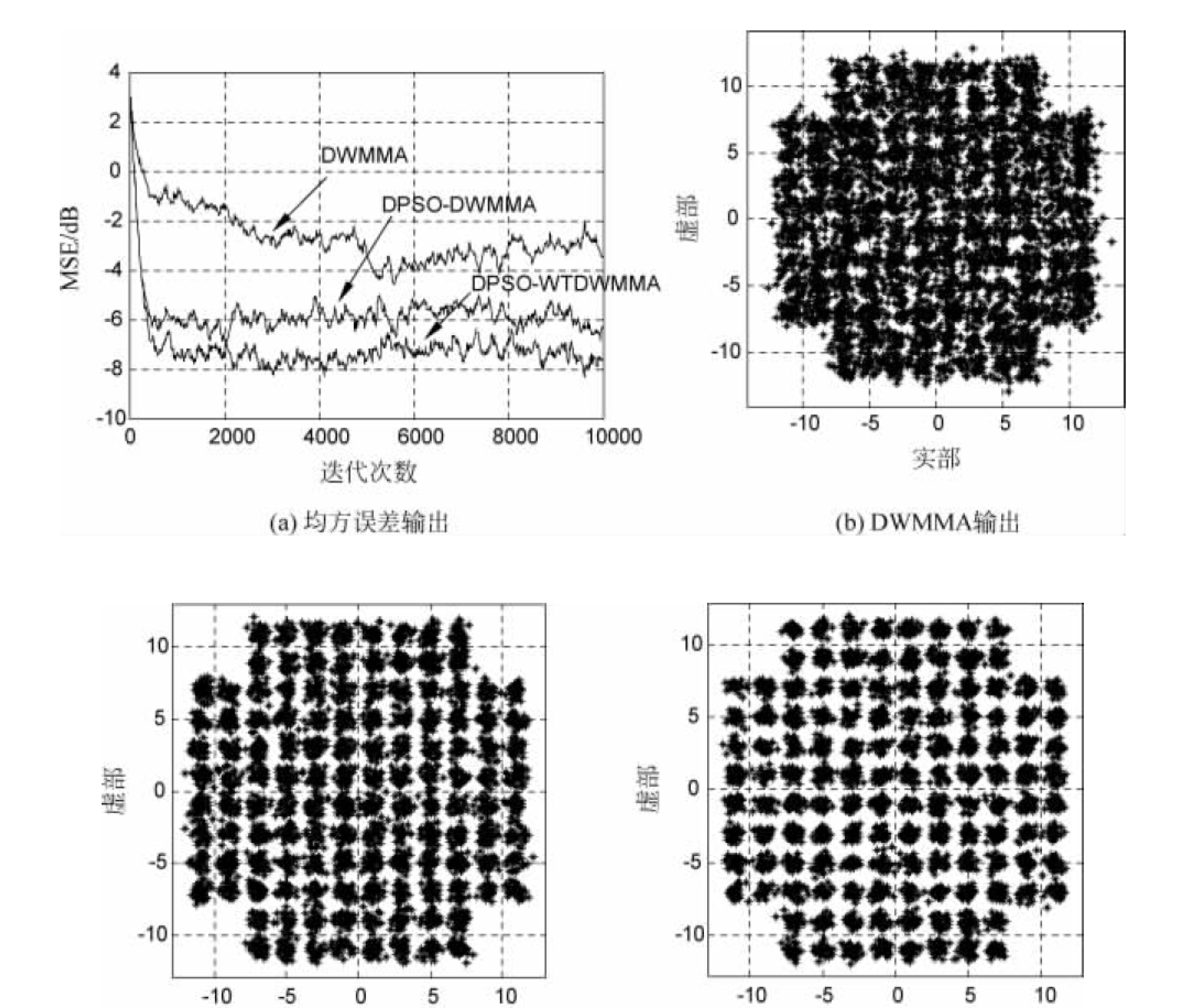



图 5.21 仿真结果

图 5.21 表明，DPSO-WTDWMMA 的收敛速度最快，比 DWMMA 快约 4000 步、与 DPSO-DWMMA 的收敛速度一致；DPSO-WTDWMMA 的稳态均方误差最小约为 －8dB，DPSO-DWMMA 收敛后的均方误差为 －6dB，DWMMA 为 －4dB；DPSO-WTDWMMA 的输出星座图比 WMMA、DPSO-WMMA 更为清晰、紧凑。因此，引入动态粒子群优化算法、正交小波变换与动态加权多模算法后，算法性能得到较大提高，可以正确地恢复传输信号，具有一定的实用性。

## 参考文献


[1] Gamot R M, Mesa A. Particle swarm optimization-tabu search approach to constrained engineering optimization problems[J]. WSEAS Transactions on Mathematics. 2008,7(11): 666-675.

[2] Sedighizadeh D, Masehian E. Particle swarm optimization methods, taxonomy and applications[J]. International Journal of Computer Theory and Engineering, 2009,5(1): 486-501.

[3] Zhan Z H, Zhang J, Li Y, et al. Adaptive particle swarm optimization[J]. IEEE Transactions on Systems Man, and Cybernetics—Part B: Cybernetics. 2009, 39(6): 1362-1381.

[4] 林川，冯全源. 基于粒子群优化算法思想的组合自适应滤波算法[J]. 电子与信息学报，2009，31(5): 1245-1248.

[5] 吕强，刘世荣. 一种信息充分交流的粒子群优化算法[J]. 电子学报，2010，38(3): 664-667.

[6] Praveen Kumar Tripathi, Sanghamitra Bandyopadhyay, Sankar Kumar Pal. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients[J]. Information Sciences. 2007. 177(22): 5033-5049.

[7] 唐忠. 粒子群算法惯性权重的研究[J]. 广西大学学报：自然科学版，2009，34(5): 640-644.

[8] 王俊伟，汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报，2005，20(2): 194-198.

[9] 胡苓苓. 基于粒子群优化算法的小波盲均衡算法仿真研究[D]. 淮南：安徽理工

大学,2012.

[10] 胡苓苓,郭业才.基于粒子群优化的正交小波盲均衡算法[J].电子与信息学报,2011,33(5):1253-1257.

[11] Özen A, Kaya I, Soysal B. Design of a fuzzy based outer loop controller for improving the training performance of LMS algorithm [C]. In Third international conference on intelligent computing. ICIC 2007, August 21-24. Qingdao, China. Vol. 2: 1051-1063.

[12] 郭业才,王丽华.模糊神经网络控制的混合小波神经网络盲均衡算法[J].电子学报,2011,39(4):975-980.

[13] 郭业才,刘振兴.基于平衡正交多小波变换的盲均衡算法[J].兵工学报,2010,31(3):279-284.

[14] 杨超.基于小波变换的联合盲均衡算法[D].淮南:安徽理工大学.2009.

[15] 王军锋,宋国乡.小波变换的自适应均衡算法[J].西安电子科技大学学报,2002,27(2):21-24.

[16] 冯文江,任志勇.误差反馈的正交小波均衡器结构与算法研究[J].系统仿真学报,2007,19(20):4762-4772.

[17] 韩迎鸽,郭业才,李保坤,等.引入动量项的正交小波变换盲均衡算法[J].系统仿真学报,2008,20(6):1559-1562.

[18] Guo Yecai, Hu Lingling. An orthogonal Wavelet Transform Blind Equalization Algorithm Based on the Optimization of Immune Clone Particle Swarm[C]. International Conference, ICICIS 2011, 2011.

[19] 魏建香,孙越泓,苏新宁.一种基于免疫选择的粒子群优化算法[J].南京大学学报(自然科学),2010,46(1):1-9.

[20] 高鹰,谢胜利.免疫粒子群优化算法[J].计算机工程与应用,2004,6:4-6.

[21] 于宗艳,韩连涛.免疫粒子群优化算法及应用[J].计算机仿真,2008,25(12):208-211.

[22] Saced Farzi, Islamic Azad. Discrete quantum-behaved particle swarm optimization for the multi-unit combinatorial auction winner determination problem[J]. Journal of Applied Sciences, 2010, 10(4): 291-297.

[23] Yecai GUO, Xueqing ZHAO, Zhenxin LIU, et al. A Modified T/2 Fractionally Spaced Coordinate Transformation Blind Equalization Algorithm[J]. Int. J. Communications. Network and System Sciences. 2010, 3: 183-189.

[24] 郭业才,丁雪洁,郭福东.正交小波分数间隔频率分集自优化盲均衡算法[J].兵工学报,2010,31(3):285-290.

[25] 胡苓苓,郭业才. 基于 QPSO 的小波分数间隔盲均衡算法[J]. 计算机工程,2011,37(24):195-198.

[26] 张路平,王建新. MQAM 信号调制方式盲识别[J]. 电子与信息学报,2011,33(2):332-336.

[27] 詹亚锋,曹志刚,马正新. M-QAM 信号的调制制式识别[J]. 通信学报,2004,25(2):68-74.

[28] 胡苓苓,郭业才. 基于量子粒子群优化的正交小波加权多模盲均衡算法[J]. 物理学报,2012,61(5):054304.

[29] Chisheng Li, Wenjuan Dong, Guojin Wan, et al. A new nultimodulus blind equalization algorithm for QAM signals[C]. 2009 5th International Conference on wireless Communications, Networking and Mobile Computing, 24-26 Sept. 2009. Beijing, China. IEEE Press, 2009: 1-4.

[30] 窦高奇,高俊. 适用于高阶 QAM 系统的多模盲均衡新算法[J]. 电子与信息学报,2008,30(2):387-391.

[31] 吴迪,葛临东,王彬. 适用于高阶 QAM 信号的混合型盲均衡算法[J]. 信息工程大学学报,2010,11(1):45-48.

[32] 徐小东,戴佩初,徐晓霞. 适合高阶 QAM 信号的加权多模盲均衡算法[J]. 电子与信息学报,2007,29(6):1352-1355.

[33] 唐英干,崔玉红,关新平. 动态粒子群算法在经济负荷分配中的应用[J]. 计算机仿真,2009,26(8):242-245.

[34] 于雪晶,麻肖妃,夏斌. 动态粒子群优化算法[J]. 计算机工程,2010,36(4):193-197.

[35] 张玉芳,薛青松,熊忠阳. 基于禁忌搜索的动态粒子群算法[J]. 计算机工程与应用,2008,44(24):56-58.

[36] 胡苓苓,郭业才. 基于动态粒子群小波动态加权多模盲均衡算法[J]. 兵工学报,2013,34(12):1548-1556.

# 第 6 章　基于人工鱼群算法优化的盲均衡算法

**【内容导读】**　本章首先分析人工鱼群行为和算法步骤，将模拟退火与人工鱼群变异相结合，优化了正交小波分数间隔盲均衡算法；利用信号星座图的先验信息，将混沌人工鱼群算法融入至广义多模盲均衡算法中，优化了基于混沌人工鱼群算法优化的广义多模盲均衡算法；将人工免疫算法与人工鱼群算法相结合，研究了正交小波频域多模盲均衡算法；提出了量子人工鱼群的概念，研究了基于量子人工鱼群算法优化的超指数自适应最小熵盲均衡算法。

## 6.1　人工鱼群算法

人工鱼群算法（artificial fish swarm algorithm，AFSA）是由李晓磊博士于 2002 年基于鱼群的活动特征提出的一种群体智能寻优算法[1-3]。在一

片水域中，一般营养物质最丰富的地方，所聚集的鱼的数目也最多。根据这一特征来模仿天然鱼的觅食、聚群、追尾和随机等行为，从而实现全局寻优搜索，这就是人工鱼群算法的基本思想。在人工鱼群算法中，人工鱼是天然鱼个体的虚拟实体，它携带了大量自身数据信息和各种行为，如图 6.1 所示。人工鱼当前的状态主要由问题的解向量空间和其他人工鱼的状态构成，它在下一时刻的行为取决于当前的自身状态和其他人工鱼的当前状态，并且人工鱼通过自身活动来影响环境，与环境实现交互式互动，从而影响其他人工鱼的活动。




图 6.1　人工鱼虚拟实体

### 6.1.1　人工鱼群行为描述

人工鱼群算法采用自下而上的设计模式，用 MATLAB 语言形式来说明。

1. 人工鱼的状态描述与参数

在人工鱼群算法中，向量 $\boldsymbol{X}_{\mathrm{AFS}}=(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_N)$ 用来表示人工鱼群的位置向量，其中 $\boldsymbol{X}_i(i=1,2,\cdots,N)$ 为第 $i$ 条人工鱼的位置向量；第 $k$ 时刻第 $i$ 条人工鱼位置向量的食物浓度为 $Y_i(k)=J(\boldsymbol{X}_i(k))$，也称 $Y$ 为目标函数。人工鱼个体之间的距离为 $d_{i,j}=\|\boldsymbol{X}_i-\boldsymbol{X}_j\|$，$\boldsymbol{X}_i$ 为第 $i$ 条人工鱼的位置向

量，$\boldsymbol{X}_j$ 为第 $j$ 条人工鱼的位置向量，$Y_i$ 为第 $i$ 条人工鱼的目标函数值，$Y_j$ 表示第 $j$ 条人工鱼的目标函数值，$Y_c$ 为中心位置的目标函数值；Visual、Step、$\delta$、try_number 分别表示人工鱼的视野范围、步长、拥挤度因子和尝试次数，如图 6.2 所示。

2．觅食行为

人工鱼当前 $k$ 时刻的位置向量 $\boldsymbol{X}_i(k)$，在视野 Visual 范围内随机选择一个位置，该位置对应的位置向量为 $\boldsymbol{X}_j(k)$，如果目标函数值 $J(\boldsymbol{X}_i(k)) > J(\boldsymbol{X}_j(k))$，则人工鱼按式(6.1.1)向位置向量 $\boldsymbol{X}_j$ 处移动一步；反之，再在视野 Visual 范围内重新随机选择位置向量 $\boldsymbol{X}_j(k)$，若多次尝试之后，仍无法达到觅食条件，则按式(6.1.2)随机移动一步。移动后的位置向量为

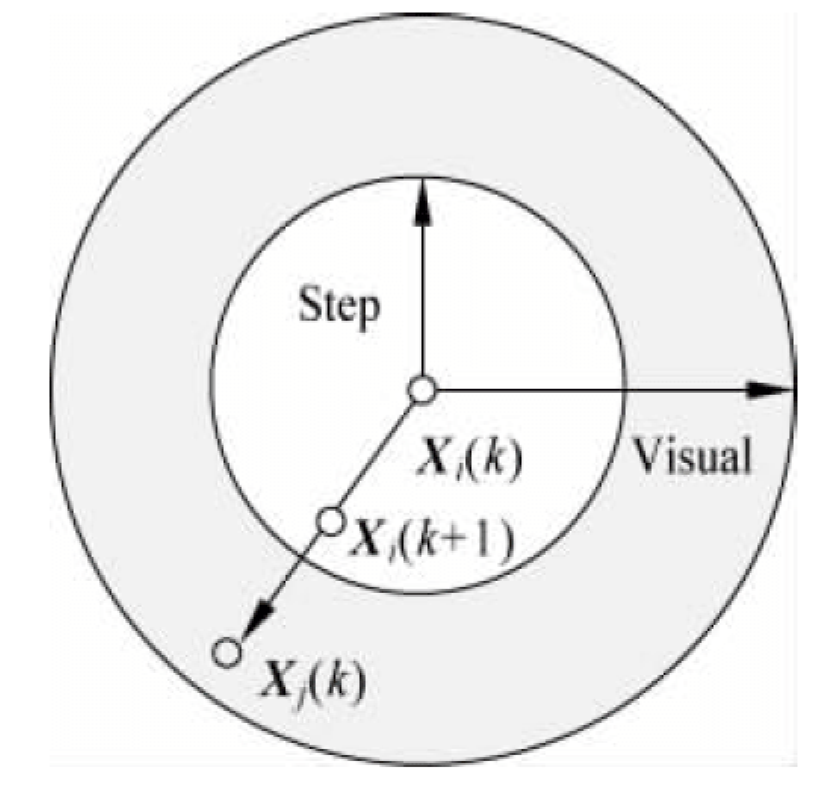



图 6.2　人工鱼概念图

$$\boldsymbol{X}_i(k+1) = \boldsymbol{X}_i(k) + \frac{\boldsymbol{X}_j(k) - \boldsymbol{X}_i(k)}{\|\boldsymbol{X}_j(k) - \boldsymbol{X}_i(k)\|} \cdot \text{step} \cdot \text{rand}(0,1) \tag{6.1.1}$$

$$\boldsymbol{X}_i(k+1) = \boldsymbol{X}_i(k) + \text{Visual} \cdot \text{rand}(0,1) \tag{6.1.2}$$

其行为代码描述如下：

```
Function [y,x] = prey(x,try_number,step,p1)
y = food(x,p1);
For j = 1:try_number
Xx = x + rand () * 2;
If norm (xx - x) < 3&&food (xx,p1) < food(x,p1)
x = x + rand * step * (xx - x) /norm (xx - x);
y = food(x,p1),break;
End
End
```

3．聚群行为

人工鱼当前 $k$ 时刻的位置向量 $\boldsymbol{X}_i(k)$，探索当前视野 Visual 范围内

$(d_{i,j}<\text{Visual})$伙伴数目 $N_f$ 及视野中心位置向量 $\boldsymbol{X}_c(k)$。视野中心位置向量为

$$\boldsymbol{X}_c(k)=\sum_{j=1}^{N_f}\boldsymbol{X}_j(k)/N_f \tag{6.1.3}$$

如果 $Y_c(k)/N_f<\delta Y_i(k)$，则表明视野中心有较多食物且不太拥挤，于是按式(6.1.4)向中心位置向移动一步；若不满足聚群条件，则进行觅食行为。

$$\boldsymbol{X}_i(k+1)=\boldsymbol{X}_i(k)+\frac{\boldsymbol{X}_c(k)-\boldsymbol{X}_i(k)}{\|\boldsymbol{X}_c(k)-\boldsymbol{X}_i(k)\|}\cdot \text{step}\cdot \text{rand}(0,1) \tag{6.1.4}$$

```
Function [y,x] = swarm(x,af,step,try_number,m,p1,p)
Xx = [];
yy = [];
y = food(x,p1);
z = [];
nf = [];
delt = 0.618;
b = [];
For i = 1: m
     If norm(x - af(i,:))<3
          b = [b;af(i,:)];
End
End
Xx = sum (b)/size (b,1);
Yy = food (xx,p1);
If yy < food(x,p1)
     NF = 0;
     For i = 1: m
     If norm(xx - af(i,:))<3
          NF = NF + 1;
     End
End
If yy /NF < delt * food(x,p (I,:))
     x = x + rand * step * (xx - x) /norm (xx - x);
End
     y = food(x,p1);
Else prey(x,try_number,step,p1);
End
```

4. 追尾行为

人工鱼当前 $k$ 时刻的位置向量 $\boldsymbol{X}_i(k)$，探索当前视野 Visual 范围内的伙伴中目标函数最大值对应的位置向量为 $\boldsymbol{X}_{max}$。如果 $Y_{max}(k)/N_f > \delta Y_i(k)$，则人工鱼向 $\boldsymbol{X}_{max}$ 移动一步；若不满足条件，则人工鱼进行聚群行为。

$$\boldsymbol{X}_i(k+1) = \boldsymbol{X}_i(k) + \frac{\boldsymbol{X}_{max}(k) - \boldsymbol{X}_i(k)}{\|\boldsymbol{X}_{max}(k) - \boldsymbol{X}_i(k)\|} \cdot \text{step} \cdot \text{rand}(0,1) \tag{6.1.5}$$

```
function[y,x] = follow(x,af,step,try_number,m,p1,pp)
Xx = [];
Yy = 100;
z = [];
y = food(x,p1);
delt = 0.618;
For i = 1: m
        if norm(x - af(i,:))< 5&&food(af(i,:),pp(i,:))< yy
        yy = food(af(i,:),pp(i,:));
        Xx = af(i,:);
        z = i;
       End
End
If yy < food(x,p1)
    Nf = 0;
    For i = 1: m
          If norm (af(z,:) - af(i,:))< 3
            Nf = nf + 1;
          End
    End
    If yy /nf < delt * food(x,p1)
        x = x + rand * step * (xx - x) /norm (xx - x);
        y = food(x,p1);
    Else prey(x,try_number,step,p1);
    End
Else prey(x,try_number,step,p1);
End
```

5. 随机行为

人工鱼的随机行为是在视野范围内随机选择一个状态，再向该方向移

动一步。其目的是在更大范围内寻找同伴或食物,它是觅食行为的一个默认行为。

6. 行为选择

根据所要解决的实际问题,评价人工鱼个体所处的环境,选择一种合适的行为。通常人工鱼的选择行为是使其能够进入下一个状态的行为,否则采取随机行为。通常,有两种方式可获得最优值:①在算法迭代结束时,最优值的分布由人工鱼的最后分布情况决定,因为人工鱼个体一般较多地聚集在全局最优值处;②在寻优迭代中,跟踪记录最优个体的状态。

### 6.1.2 人工鱼群算法步骤

人工鱼群算法在搜索前期收敛速度较快,在后期存在计算精度低、易于陷入局部极值等缺点。影响搜索能力的主要因素有:尝试次数、拥挤度因子、步长、视野。人工鱼群寻优速度大部分取决于这4种参数。因此,参数设置对人工鱼群算法性能有重要影响,是人工鱼群算法优化过程的重要部分之一。

人工鱼群优化算法的具体流程,如图6.3所示。

人工鱼群算法步骤如下。

步骤1:人工鱼群种群初始化及参数设置。设人工鱼群初始种群$\boldsymbol{X}_{\mathrm{AFS}}=(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_N)$,其中$\boldsymbol{X}_i(i=1,2,\cdots,N)$为第$i$条人工鱼的初始位置向量。

步骤2:计算适应度函数。计算每条人工鱼的适应度函数,将适应度最大值及其对应的位置向量分别记录在公告牌中。

步骤3:人工鱼群的行为选择。人工鱼群中每条人工鱼发生追尾行为操作,若追尾不成功,则进行聚群行为操作,若聚群行为不成功,则进行觅食行为操作,人工鱼的当前位置向量发生改变。

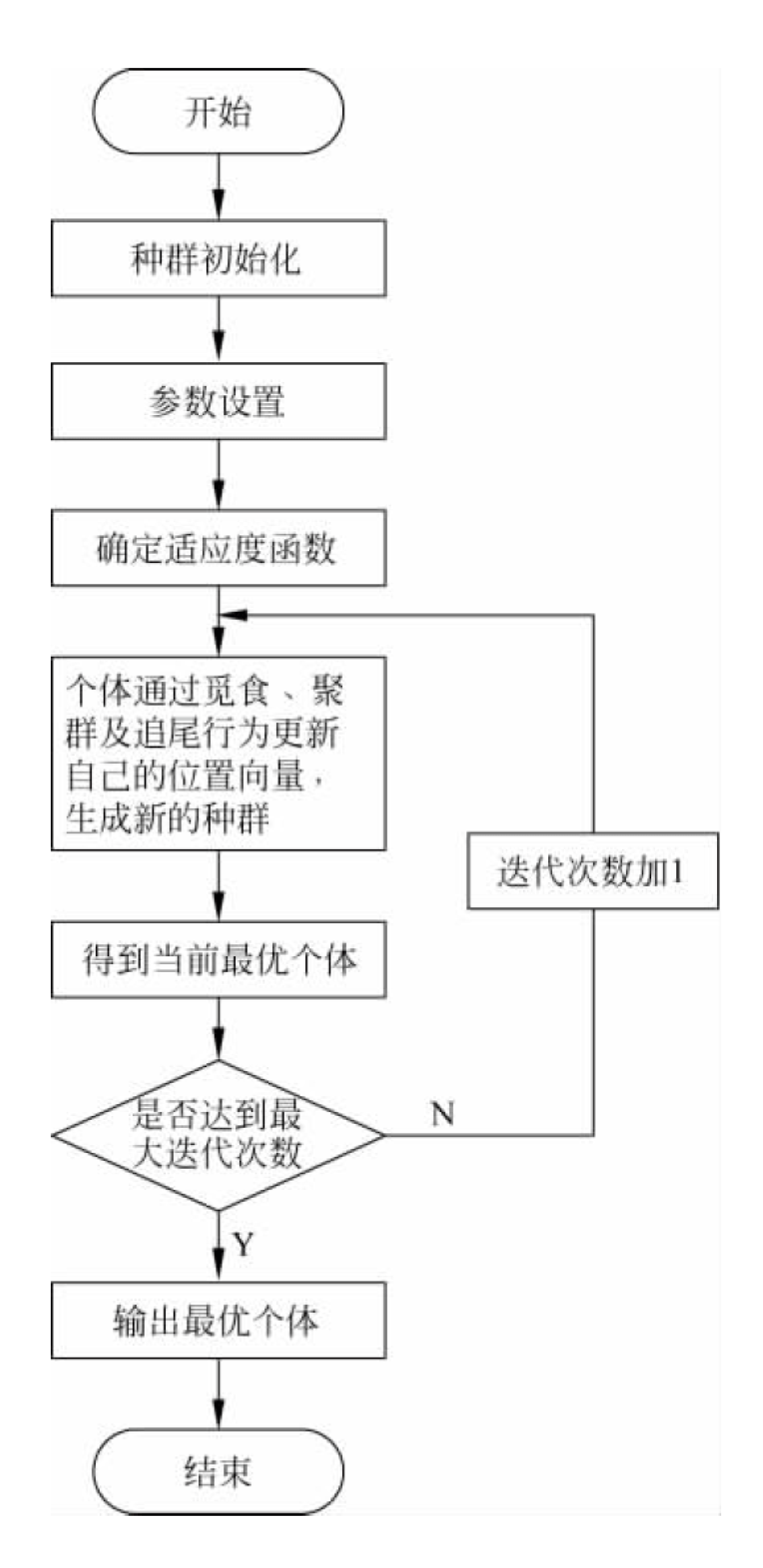



图 6.3　人工鱼群算法流程图

步骤 4：更新公告牌。

每次迭代之后，每条人工鱼的位置向量发生改变，计算每条人工鱼新位置向量对应的适应度值，将适应度最大值与公告牌中保存的适应度最大值进行比较；如果当前适应度值大，则用当前适应度最大值及其对应的人工鱼位置向量更新公告牌中内容。

步骤 5：终止条件判断。

判断人工鱼群算法的迭代次数是否达到最大迭代次数。若满足条件，

则输出公告牌中的位置向量，终止算法；若没有达到终止条件，跳转到步骤 3。

由以上流程可知，AFSA 是利用人工鱼个体的压缩与扩张，将局部和全局搜索结合起来而获得实际问题的最优解域。或者可以解释为：AFSA 将低维度空间问题转化到高维度空间中去求解，再将解的结果投影到低维度空间中，从而获得待优化问题的满意解。

## 6.2 基于模拟退火与人工鱼群变异混合优化的正交小波盲均衡算法

人工鱼群算法是一种基于动物行为的群体智能全局寻优算法，具有较强的全局寻优能力，但有以下两个缺点[4]：①当寻优区域较大且处于变化并不激烈时，收敛于全局极值的速度慢，搜索能力下降；②容易出现“早熟”现象。

本节有机融合了变异算子、模拟退火算法（Simulated Annealing，SA）[5-6]、人工鱼群算法及分数间隔正交小波盲均衡算法，研究了一种基于模拟退火与人工鱼群变异混合优化的正交小波盲均衡算法[7]。该算法分为两个过程。

第一个过程：全局搜索。由分数间隔正交小波盲均衡算法的代价函数定义人工鱼群算法的适应度函数，利用人工鱼群算法快速寻找均衡器的最优权向量，并在寻优过程中引入变异算子，改变部分人工鱼的状态，保持人工鱼种群的多样性、加快收敛速度。

第二个过程：局部搜索。通过第一个过程得到最优值后，用 SA 算法对人工鱼最优个体进行“精细搜索”局部优化，获得高精度的最优值。

### 6.2.1 算法原理

基于模拟退火与人工鱼群变异混合优化的分数间隔正交小波盲均衡

算法原理如图 6.4 所示。

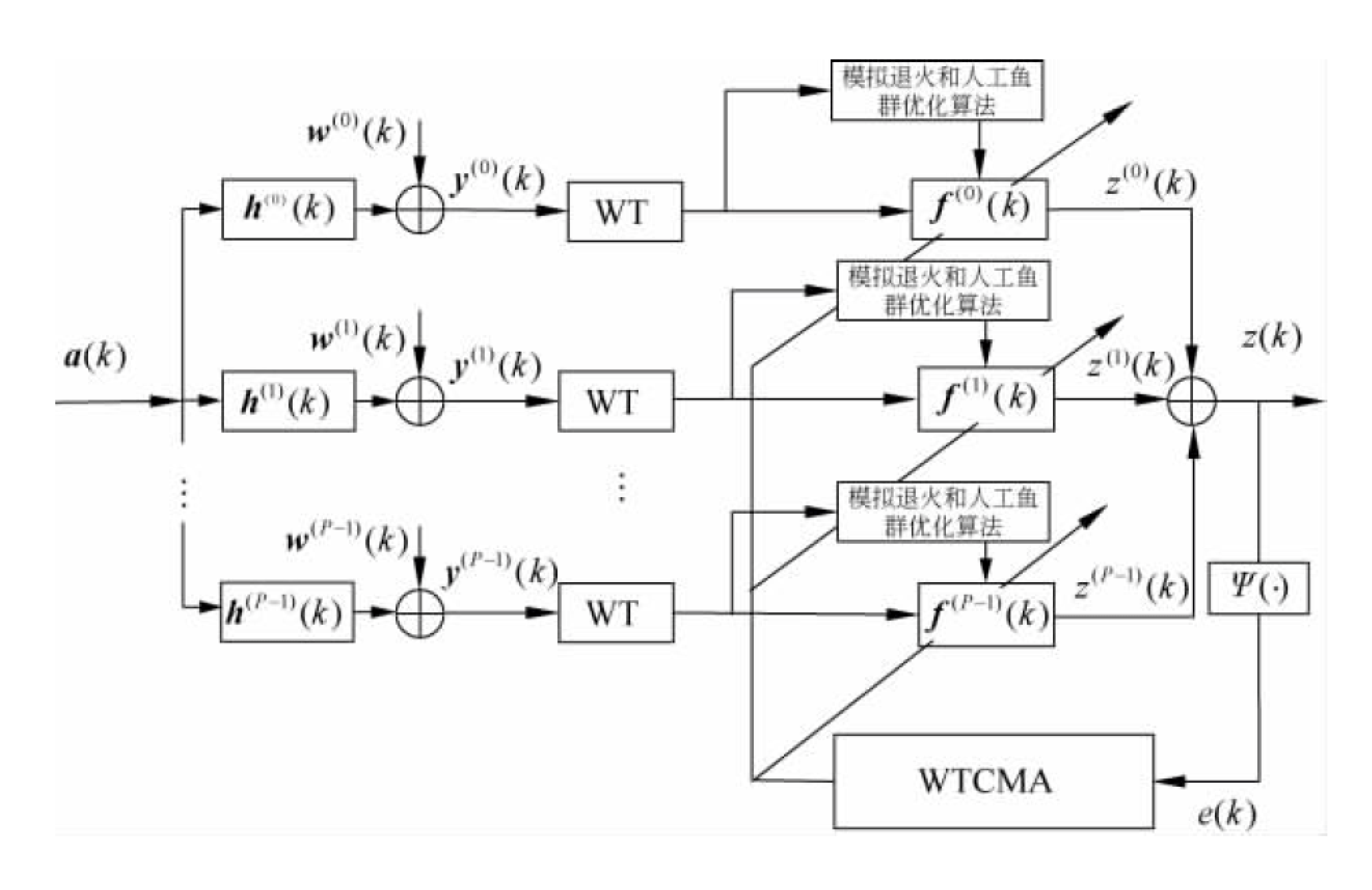



图 6.4 基于 SA 与 AFSA 变异混合优化的正交小波分数间隔盲均衡算法

发射序列 $\boldsymbol{a}(k)$为独立同分布的 QAM 或 PSK 调制信号。我们假设发射信号序列的最大值为

$$\max|\mathrm{Re}\{\boldsymbol{a}(k)\}| = \max|\mathrm{Im}\{\boldsymbol{a}(k)\}| = \gamma \tag{6.2.1}$$

假设 $H^{(p)}(z)=\sum h^{(p)}(k)z^{-k}, p=0,1,\cdots,P-1$，子均衡器 $F^{(p)}(z)=\sum f^{(p)}(k)z^{-k}$。第 $p$ 个子信道的输入输出关系为

$$\boldsymbol{y}^{(p)}(k)=\boldsymbol{a}(k)\boldsymbol{h}^{(p)}(k)+\boldsymbol{w}^{(p)}(k) \tag{6.2.2}$$

均衡器的输出为

$$z(k)=\sum_{p=0}^{P-1}z^{(p)}(k)=\sum_{p=0}^{P-1}\boldsymbol{y}^{(p)}(k)\boldsymbol{f}^{(p)}(k) \tag{6.2.3}$$

若在所有子信道中，最高阶数为 $L_{\max}$，且每一路子信道的均衡器阶数为 $L_f$。因此，可将式(6.2.2)和式(6.2.3)写成矩阵形式，即

$$\boldsymbol{y}(k)=\boldsymbol{H}\boldsymbol{a}(k)+\boldsymbol{w}(k) \tag{6.2.4}$$

$$z(k)=\boldsymbol{f}^{\mathrm{H}}\boldsymbol{y}(k) \tag{6.2.5}$$

式中

$$\boldsymbol{a}(k)=[a(k),a(k-1),\cdots,a(k-M-L_{\max})]^{\mathrm{T}} \tag{6.2.6}$$

$$\boldsymbol{w}(k)=[w(k),\cdots,w(k-M)]^{\mathrm{T}} \tag{6.2.7}$$

$$\boldsymbol{y}(k)=[y^{(0)}(k),\cdots,y^{(0)}(k-N),\cdots,y^{(P-1)}(k),\cdots,y^{(P-1)}(k-L_f)]^{\mathrm{T}} \tag{6.2.8}$$

$$\boldsymbol{f}=[f^{(0)}(k),\cdots,f^{(0)}(L_f),\cdots,f^{(P-1)}(0),\cdots,f^{(P-1)}(L_f)]^{\mathrm{T}} \tag{6.2.9}$$

$$\boldsymbol{H}^{(p)}=\begin{bmatrix} h^{(p)}(0) & \cdots & h^{(p)}(L_{\max}) & 0 & \cdots & 0 \\ 0 & h^{(p)}(0) & \cdots & h^{(p)}(L_{\max}) & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h^{(p)}(0) & \cdots & h^{(p)}(L_{\max}) \end{bmatrix}_{(L_f+1)\times(L_f+L_{\max}+1)} \tag{6.2.10}$$

在有限长度下，每个子信道的均衡器权系数 $f^{(p)}(k)$ 为

$$f^{(p)}(k)=\sum_{j=1}^{J}\sum_{m=0}^{k_j}d_{jm}^{(p)}\cdot\varphi_{jm}^{(p)}(k)+\sum_{m=0}^{k_j}v_{Jm}^{(p)}\cdot\phi_{Jm}^{(p)}(k) \tag{6.2.11}$$

式中，$\varphi_{jm}^{(p)}(k)$ 和 $\phi_{Jm}^{(p)}(k)$ 分别表示尺度函数和小波函数；$k_j=L_f/2^J-1(j=1,2,\cdots,J)$，$J$ 为小波分解的最大尺度；$d_{jm}^{(p)}$ 和 $v_{Jm}^{(p)}$ 分别为每路均衡器的权系数。

每路子信道均衡器的输出为

$$\begin{aligned} z^{(p)}(k) &= \sum_{n=0}^{L_f-1}\boldsymbol{f}_n^{(p)}(k)\cdot\boldsymbol{y}^{(p)}(k-n) \\ &= \sum_{j=1}^{J}\sum_{m=0}^{k_j}d_{jm}^{(p)}(k)\cdot u_{jm}^{(p)}(k)+\sum_{m=0}^{k_j}v_{Jm}^{(p)}(k)\cdot s_{Jm}^{(p)}(k) \end{aligned} \tag{6.2.12}$$

式(6.2.12)表明，$\boldsymbol{f}^{(p)}(k)$ 用小波基函数表示的实质是，对每路子信道均衡器的输入信号进行小波变换，从而改变均衡器的结构。设 $\boldsymbol{R}^{(p)}(k)=[u_{1,0}^{(p)}(k),\cdots,u_{1,k_j}^{(p)}(k),s_{J,0}^{(p)}(k),\cdots,s_{J,k_j}^{(p)}(k)]^{\mathrm{T}}$，$\boldsymbol{f}^{(p)}(k)=[d_{1,0}^{(p)}(k),\cdots,d_{J,k_j}^{(p)}(k),v_{J,0}^{(p)}(k),\cdots,v_{J,k_j}^{(p)}(k)]^{\mathrm{T}}$，则

$$\boldsymbol{R}^{(p)}(k)=\boldsymbol{y}^{(p)}(k)\boldsymbol{V} \tag{6.2.13}$$

$$z^{(p)}(k)=\boldsymbol{f}^{(p)\mathrm{H}}(k)\boldsymbol{R}^{(p)}(k) \tag{6.2.14}$$

这时，子均衡器权向量的迭代公式为

$$f^{(p)}(k+1)=f^{(p)}(k)+\mu[\hat{R}^{(p)}(k)]^{-1}e(k)R^{(p)}(k)z^{(p)}(k) \qquad (6.2.15)$$

式中，$[\hat{R}^{(p)}(k)]^{-1}=\mathrm{diag}[\sigma_{1,0}^{2(p)}(k),\sigma_{1,1}^{2(p)}(k),\cdots,\sigma_{J,k_j-1}^{2(p)}(k),\sigma_{J+1,0}^{2(p)}(k),\cdots,\sigma_{J+1,k_j-1}^{2(l)}(k)]$。

$$\sigma_{j,m}^{2(p)}(k+1)=\beta\sigma_{j,m}^{2(p)}(k)+(1-\beta)\mid u_{j,m}^{(p)}(k)\mid^2 \qquad (6.2.16)$$

$$\sigma_{J+1,m}^{2(p)}(k+1)=\beta\sigma_{J,m}^{2(p)}(k)+(1-\beta)\mid s_{J,m}^{(p)}(k)\mid^2 \qquad (6.2.17)$$

式中，$\beta$ 为遗忘因子，$\sigma_{j,m}^2(k)$ 和 $\sigma_{J,m}^2(k)$ 分别表示小波变换系数 $u_{j,m}(k)$ 和尺度变换系数 $s_{n,m}(k)$ 的平均功率估计。信号经正交小波变换，并进行能量归一化处理，得到分数间隔正交小波盲均衡算法（FSE-WTCMA）[8]。该算法缺乏全局搜索能力，同时代价函数需满足绝对可导。将具有优良局部寻优能力的 SA 算法与全局搜索能力强的带变异算子的 AFSA 有机结合，用于优化 FSE-WTCMA。其优化流程如图 6.5 所示。

1. 确定适应度函数

适应度函数一般是通过代价函数转化而来的。WTCMA 将代价函数经多次迭代而找到极小值，此时对应的均衡器权向量是最优的[9]。人工鱼群算法的适应度函数定义为

$$\mathrm{Fit}(f_i)=\frac{1}{J_{\text{FSE-WTCMA}}(f_i)},\quad i=1,2,\cdots,N \qquad (6.2.18)$$

式中，$J_{\text{FSE-WTCMA}}$ 是均衡器的代价函数，$f_i$ 是 AFSA 寻找到的均衡器权向量。

2. 设计变异算子

由于 AFSA 在后期会出现精度降低、收敛速度减慢等问题。因此，在 AFSA 中嵌入变异算子，对人工鱼进行变异操作以保持人工鱼种群的多样性、提高搜索精度和速度。随着进化次数的增加，判断搜索结果是否有改进，可在算法中设立公告板来记录寻优历史中最优人工鱼状态。若在多次进化过程中，公告板变化很小甚至没有变化时，则对人工鱼群中除公告板外的所有人工鱼执行变异操作。其变异操作的设计如下：

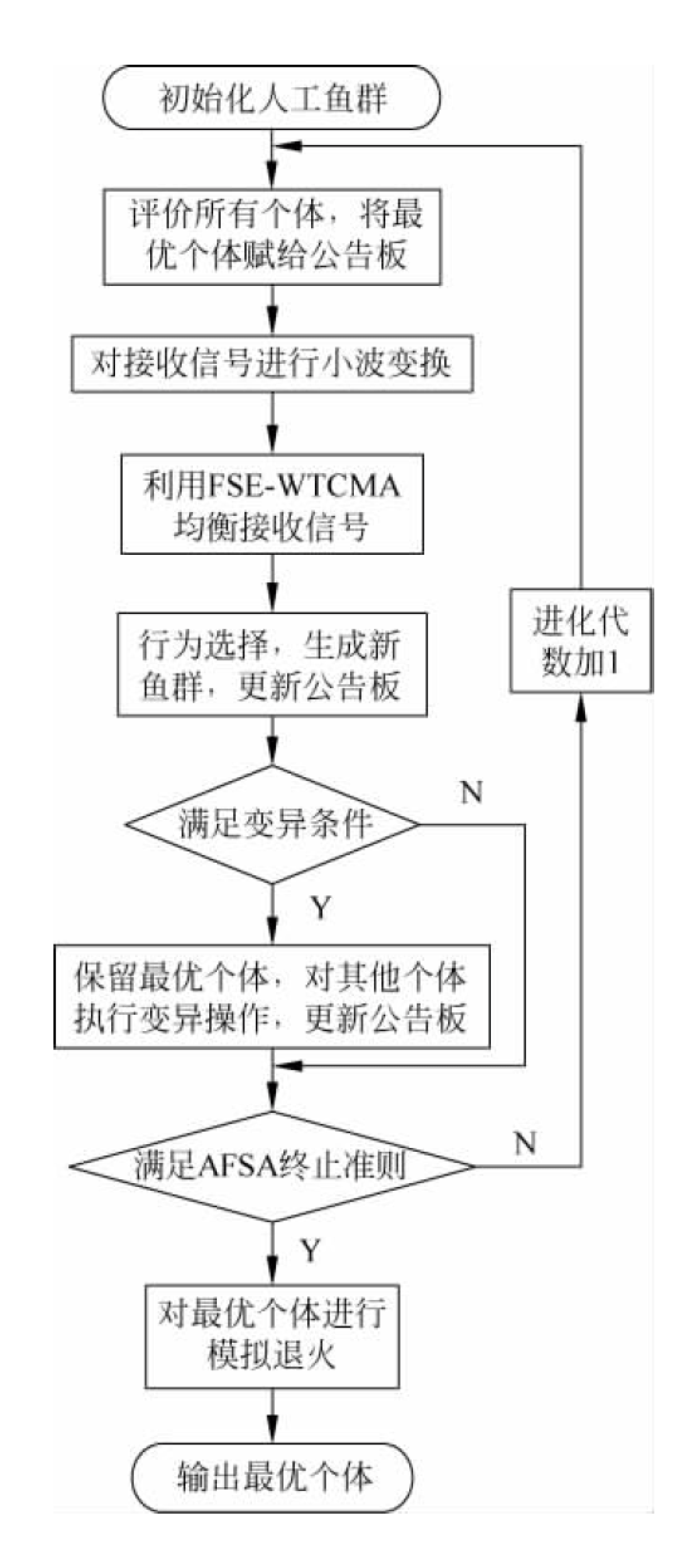



图 6.5　模拟退火与人工鱼群变异混合算法流程

(1) 对各人工鱼每一维产生一个随机数 rand∈(0,1),设置变异概率的阈值 $u_m$。

(2) 若 rand≤$u_m$,对该人工鱼个体的这一维数字随机初始化,否则保持该维数字不变。

(3) 将新生成的人工鱼个体适应度值与公告板比较,若优于公告板,更新公告板；否则,保持不变。

### 3. 模拟退火操作

由于带变异算子的 AFSA 局部搜索能力不强，故搜索到的最优解精度不高。而 SA 具有很强的局部搜索能力，因此，将由带变异算子的 AFSA 得到的最优解按 Metropolis 准则来弃留。在 SA 的每次迭代过程中，若最优解的适应度值增大，那么它被接受，否则须由式(6.2.18)判断是否被接受。

$$Q(T(t+1)) = \begin{cases} 1, & \mathrm{Fit}(t+1) < \mathrm{Fit}(t) \\ \min\left[1, \exp\left(-\dfrac{\mathrm{Fit}(t+1)-\mathrm{Fit}(t)}{\mathrm{Fit}(t+1)}\right)\right] > \mathrm{rand}, & \mathrm{Fit}(t+1) \geqslant \mathrm{Fit}(t) \end{cases} \tag{6.2.19}$$

式中，$\mathrm{Fit}(t)$为第 $i$ 次迭代所得的适应度最优值，$Q(T(t+1))$为温度 $T(t+1)$下的接收概率，$T(t+1)$可表示为

$$T(t+1) = \alpha \cdot T(t) \tag{6.2.20}$$

式中，$\alpha$ 为温度冷却系数。

由于带变异算子的 AFSA 受到视野范围、随机步长、随机行为等环境因素的影响，使它在初始化盲均衡器权向量时，得到的最优解精度不高，因此在带变异算子的 AFSA 获得最优解之后，再级联局部搜索能力强的 SA，得到“高精度”的全局最优解。

## 6.2.2 算法仿真

现以 T/2 分数间隔盲均衡算法为例，验证 SAFSA-FSE-WTCMA 的有效性，将其与 WTCMA 和 AFSA-FSE-WTCMA 对比，进行仿真实验，并以最小均方误差(MSE)曲线作为算法的性能指标。在试验中，人工鱼群规模为 100，视野为 0.15，步长为 0.03，拥挤度因子为 0.618，最大进化次数为 100，初始温度为 $T=30$，冷却参数 $\alpha=0.89$，信噪比为 20dB。

**【实验 6.1】** 最小相位水声信道 $\boldsymbol{h}=[0.9656\quad -0.0906\quad 0.5780\quad 0.2368]$，当发射信号为 16PSK 时，均衡器权长为 16，初始功率设置为 8，FSE-WTCMA 的第 3 个抽头初始化为 1，其他参数如表 6.1 所示。在保证眼图完全清晰睁开的前提下，蒙特卡罗仿真结果如图 6.6 所示。当发射信号为 16QAM 时，均衡器权长为 16，初始功率设置为 4，FSE-WTCMA 的第 6 个抽头初始化为 1，其他参数如表 6.2 所示。在保证眼图完全清晰睁开的前提下，蒙特卡罗仿真结果如图 6.7 所示。

表 6.1　仿真参数

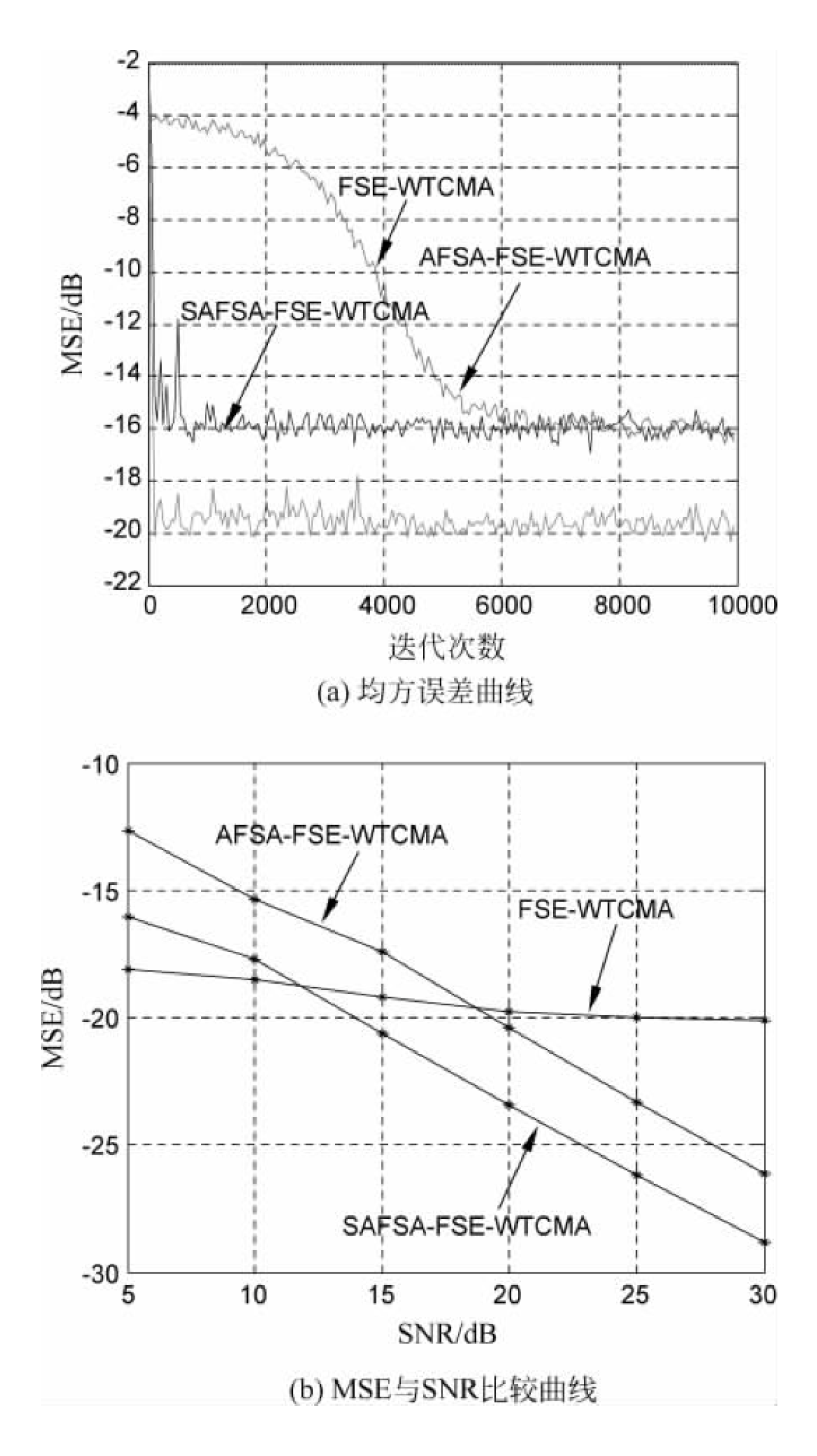
| 算　　法 | 步长 | 小波 | 分解层数 | 遗忘因子 |
|---|---|---|---|---|
| FSE-WTCMA | 0.000 04 | DB2 | 3 | 0.99 |
| AFSA-FSE-WTCMA | 0.000 05 | DB2 | 3 | 0.999 |
| SAFSA-FSE-WTCMA | 0.0009 | DB2 | 3 | 0.95 |

表 6.2　仿真参数

| 算　　法 | 步长 | 小波 | 分解层数 | 遗忘因子 |
|---|---|---|---|---|
| FSE-WTCMA | 0.000 02 | DB4 | 2 | 0.99 |
| AFSA-FSE-WTCMA | 0.000 01 | DB4 | 2 | 0.999 |
| SAFSA-FSE-WTCMA | 0.000 01 | DB4 | 2 | 0.95 |

图 6.6 表明，在收敛速度上，SAFSA-FSE-WTCMA 比 AFSA-FSE-WTCMA 快约 500 步，比 FSE-WTCMA 快 6000 步左右；在稳态误差上，SAFSA-FSE-WTCMA 与 AFSA-FSE-WTCMA 相比，减小了大约 3dB，与 FSE-WTCMA 相比，减小了大约 3dB。

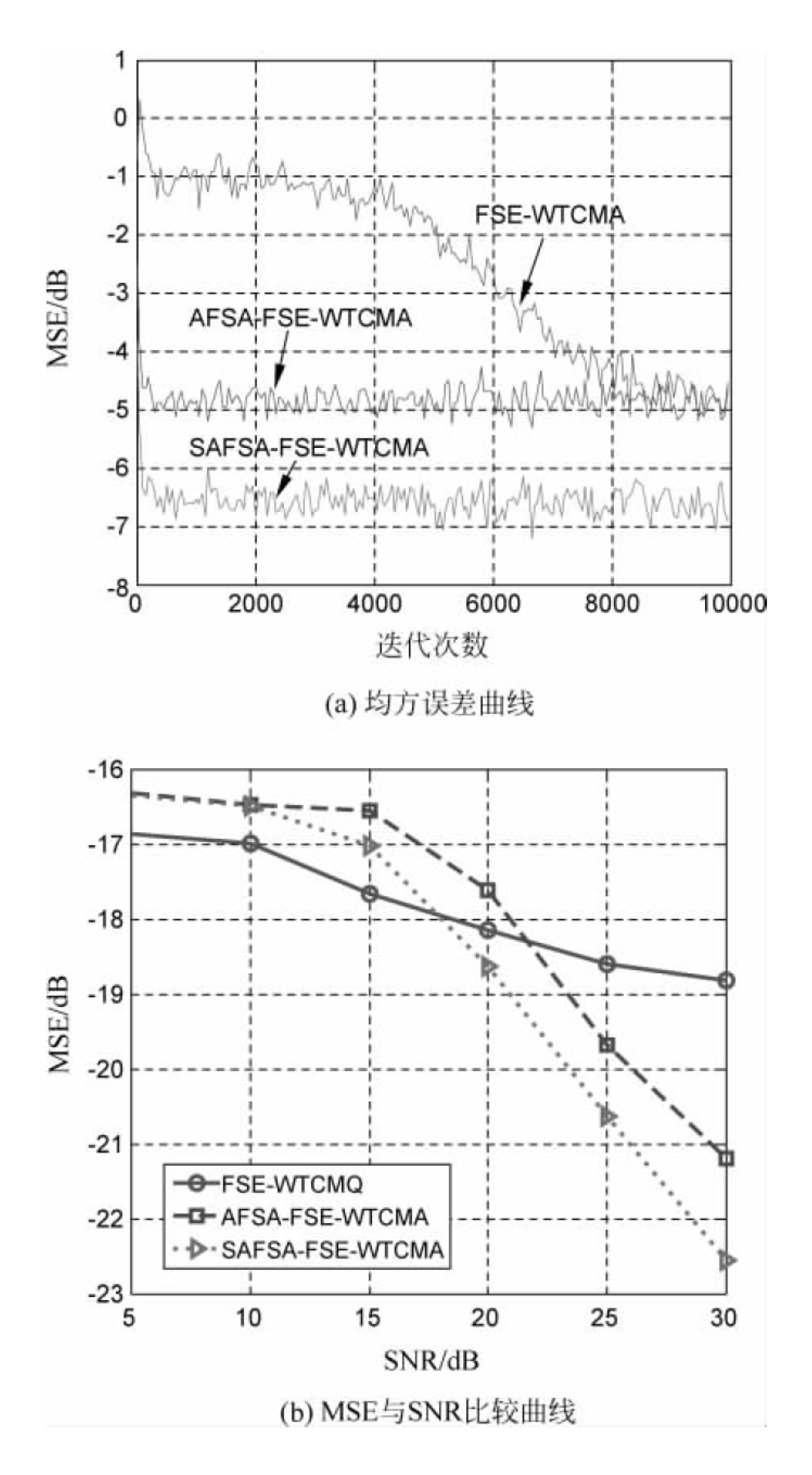
图 6.7 表明，在收敛速度上，SAFSA-FSE-WTCMA 与 AFSA-FSE-WTCMA 基本相同，比 FSE-WTCMA 快 7000 步左右；在稳态误差上，SAFSA-FSE-WTCMA 与 AFSA-FSE-WTCMA 相比，减小了约 2dB，与 FSE-WTCMA 相比，减小了约 2dB。




(a) 均方误差曲线

(b) MSE与SNR比较曲线

图 6.6　16PSK 信号的仿真结果



(a) 均方误差曲线

(b) MSE与SNR比较曲线

图 6.7　16QAM 信号的仿真结果

## 6.3 基于混沌人工鱼群算法优化的广义多模盲均衡算法

广义多模盲均衡算法(Generalized Multi-modulus Algorithm, GMMA)是将高阶 QAM 信号的星座图分成几个不同的子区域,每个子区域在权向量更新过程中通过判断输入信号所在区域来调整代价函数中的模值,并在均衡器权向量更新中自适应地修正模值。GMMA 具有比 MMA 更好的收敛性能,但收敛速度慢、稳态误差仍较大[10]。人工鱼群算法具有收敛速度快、全局搜索能力强等特点,但同时易陷入局部最优值,出现"早熟"。混沌运动能在一定范围内按照自身运动规律无重复地遍历所有状态,具有较强的局部搜索能力。因此将混沌算法嵌入至人工鱼群算法,获得混沌人工鱼群算法,并融入 GMMA 中,得到基于混沌人工鱼群算法优化的广义多模盲均衡算法(Chaotic Artificial Fish Swarm Algorithm based generalized multi-modulus algorithm, CAFSA-GMMA),有望避免人工鱼群陷入"早熟",使 GMMA 跳出局部最优值。

### 6.3.1 传统多模盲均衡算法

传统多模盲均衡算法的星座图分为正方形和十字形,正方形星座图的代价函数为[11]

$$J_{\mathrm{MMA}} = E[(z_{\mathrm{R}}^2(k) - R_{\mathrm{M}}^2)^2 + (z_{\mathrm{I}}^2(k) - R_{\mathrm{M}}^2)^2] \tag{6.3.1}$$

$$R_{\mathrm{M}}^2 = \frac{E(a^4(k) + b^4(k))}{E(a^2(k) + b^2(k))} = \frac{E(a^4(k))}{E(a^2(k))} = \frac{E(b^4(k))}{E(b^2(k))} \tag{6.3.2}$$

式中,$R_{\mathrm{M}}$ 为发射信号模值;$a(k)$与$b(k)$分别表示独立同分布发射信号的实部与虚部,故二者的期望值相等;$z_{\mathrm{R}}(k)$与$z_{\mathrm{I}}(k)$分别表示均衡器输出信号的实部与虚部。因此,式(6.3.1)可以写成实部与虚部代价函数和的形式,即

$$J_{\mathrm{R}} = E[(z_{\mathrm{R}}^2(k) - R_{\mathrm{M}}^2)^2] \tag{6.3.3}$$

$$J_{\mathrm{I}} = E[(z_{\mathrm{I}}^2(k) - R_{\mathrm{M}}^2)^2] \tag{6.3.4}$$

$$J_{\mathrm{MMA}} = J_{\mathrm{R}} + J_{\mathrm{I}} = E[(z_{\mathrm{R}}^2(k) - R_{\mathrm{M}}^2)^2] + E[(z_{\mathrm{I}}^2(k) - R_{\mathrm{M}}^2)^2] \tag{6.3.5}$$

分别对代价函数 $J_{\mathrm{R}}$ 和 $J_{\mathrm{I}}$ 求导可得,权向量更新公式为

$$\boldsymbol{f}_{\mathrm{R}}(k+1) = \boldsymbol{f}_{\mathrm{R}}(k) - \mu(z_{\mathrm{R}}^2(k) - R_{\mathrm{M}}^2)z_{\mathrm{I}}(k)\boldsymbol{y}(k) \tag{6.3.6}$$

$$\boldsymbol{f}_{\mathrm{I}}(k+1) = \boldsymbol{f}_{\mathrm{I}}(k) - \mu(z_{\mathrm{I}}^2(k) - R_{\mathrm{M}}^2)z_{\mathrm{I}}(k)\boldsymbol{y}(k) \tag{6.3.7}$$

式中,$\boldsymbol{f}_{\mathrm{R}}(k)$和 $\boldsymbol{f}_{\mathrm{I}}(k)$分别为均衡器权向量的实部和虚部;$\mu$ 为步长;$\boldsymbol{y}(k)$为均衡器输入信号。

对于十字形星座图,以 128-QAM 为例,其星座图如图 6.8 所示。图 6.8 表明,信号的实部和虚部都取自 QAM 信号的调制点集合 $\Omega\{\pm1,\pm3,\pm5,\pm7,\pm9,\pm11\}$,且每个信号出现在调制点±9,±11 的概率小于出现在±1,±3,±5,±7 的点。将正方形星座图的代价函数进行修正,得到十字形星座图的代价函数为

$$J_{\mathrm{R}} = E[(z_{\mathrm{R}}^2(k) - R_1^2)^2], \quad |z_{\mathrm{I}}(k)| < k \tag{6.3.8}$$

$$J_{\mathrm{I}} = E[(z_{\mathrm{I}}^2(k) - R_1^2)^2], \quad |z_{\mathrm{R}}(k)| < k \tag{6.3.9}$$

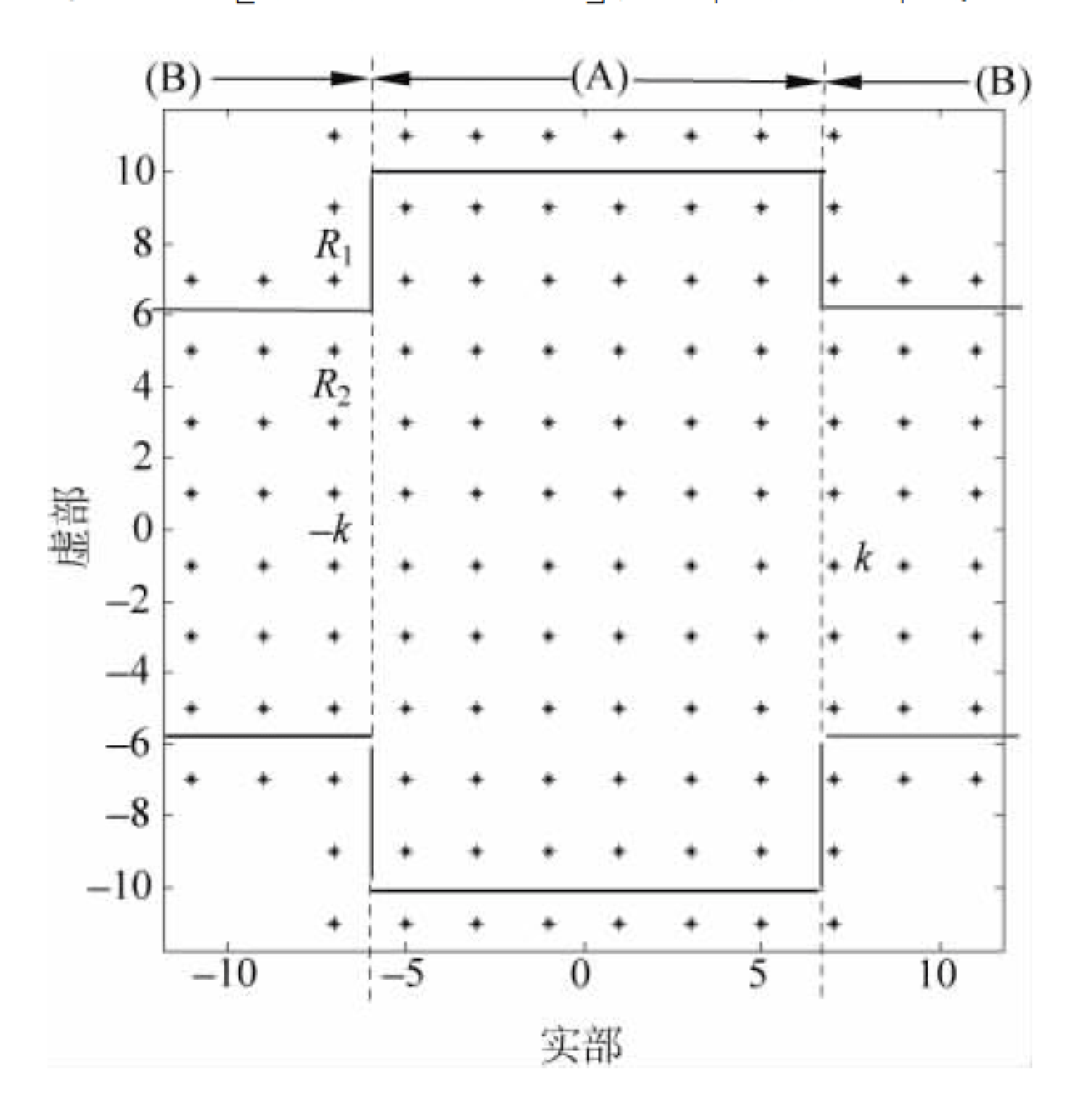



图 6.8 128-QAM 星座图

$$J_{\mathrm{R}} = E[(z_{\mathrm{R}}^2(k) - R_2^2)^2], \quad |z_{\mathrm{I}}(k)| > k \tag{6.3.10}$$

$$J_{\mathrm{I}} = E[(z_{\mathrm{I}}^2(k) - R_2^2)^2], \quad |z_{\mathrm{R}}(k)| < k \tag{6.3.11}$$

对于模值 $R_1$ 若仅考虑图 6.8 的第一象限，则 $a(k)=1,3,5,7,9,11$，$b(k)=1,3,5,7$，那么 $b(k)$ 出现的概率 $4/24=1/6$。同理，对于模值 $R_2$，$b(k)=1,3,5,7$，$a(k)=9,11$，那么 $b(k)$ 出现的概率 $2/8=1/4$。因此，在(A)区中，有

$$E[b^2(k)] = \frac{1}{6}(1^2+3^2+5^2+7^2+9^2+11^2) = 46.67 \tag{6.3.12}$$

在(B)区中，有

$$E[b^2(k)] = \frac{1}{6}(1^2+3^2+5^2+7^2) = 21 \tag{6.3.13}$$

对于 128-QAM 星座图，由式(6.3.2)、式(6.3.12)与式(6.3.13)知，$R_1 \approx 9.2$，$R_2 \approx 6.1$。

通过以上分析，我们现归纳所有阶数十字形和正方形 MMA 模值的通用计算方法。

对于多模模值，一般通过对 MMA 的代价函数进行处理而获得。MMA 的代价函数，如式(6.3.1)所示。当均衡器收敛时，输出信号 $z_{\mathrm{R}}(k) \to a(k)$，$z_{\mathrm{I}}(k) \to b(k)$。据此，式(6.3.1)变为

$$J_{\mathrm{MMA}}(k) = E[(|a(k)|^L - R_{\mathrm{M}}^L)^2 + (|b(k)|^L - R_{\mathrm{M}}^L)^2] \tag{6.3.14}$$

式中，$L$ 是一个正整数。因为 $a(k)$ 和 $b(k)$ 具有相同的统计特性，所以输出信号的同相和正交分量也具有相同的特性。所以，在下面分析中，我们仅讨论信号的同相分量。将式(6.3.1)中的代价函数对实权向量求导，得

$$\nabla J_{\mathrm{MMA}}(k) = 2L \times E[(|z(k)|^L - R_{\mathrm{M}}^L)|z(k)|^{L-2} z(k) \boldsymbol{f}(k)] \tag{6.3.15}$$

当均衡器处于理想状态时，能准确计算模值。假设处于不同区域的信号是不相关的，则有 $E[a(k)\boldsymbol{f}(k)] = \boldsymbol{K}E[a^2(k)]$，其中，$\boldsymbol{K}$ 是一个确知的矩阵，可理解为信道矩阵。因此，得

$$E[(|a(k)|^{L}-R_{M}^{L})|a(k)|^{L-2}z(k)\boldsymbol{f}(k)]=0$$
$$\boldsymbol{K}\times E[(|z(k)|^{L}-R_{M}^{L})|z(k)|^{L-2}z^{2}(k)]=0$$
$$E[|a(k)|^{2L}]-R_{M}^{L}E[|a(k)|^{L}]=0 \tag{6.3.16}$$

由式(6.3.16),得

$$R_{M}^{L}=\frac{E[a^{2L}(k)]}{E[|a(k)|^{L}]} \tag{6.3.17}$$

对于正方形星座图:通过上面分析知道,模值 $R$ 是信号 $a(k)$ 在第 $k$ 时刻的函数。假设信号取为 $\{a(k)\}=\{\pm1,\pm3,\cdots,\pm(2m+1)\}$ 时,其中 $m$ 为已调信号的最大功率幅值。参照文献[12],得

$$\sum_{K=1}^{m}K=\frac{1}{2}m(m+1) \tag{6.3.18}$$

$$\sum_{K=1}^{m}K^{2}=\frac{1}{6}m(m+1)(2m+1) \tag{6.3.19}$$

$$\sum_{K=1}^{m}K^{3}=\frac{1}{4}[m(m+1)]^{2} \tag{6.3.20}$$

$$\sum_{K=1}^{m}K^{4}=\frac{1}{30}m(m+1)(2m+1)(3m^{2}+3m-1) \tag{6.3.21}$$

上面等式不能直接用于求解信号幅值,但可获取已调信号功率的近似形式为

$$\sum_{K=1}^{m}(2K-1)=\sum_{K=1}^{2m-1}K-2\sum_{K=1}^{m-1}K=m^{2} \tag{6.3.22}$$

利用以上表达式,可将已调信号功率幅值表示为

$$\sum|a(k)|=\sum_{K=1}^{m}(2K-1)=m^{2} \tag{6.3.23}$$

$$\sum a^{2}(k)=\sum_{K=1}^{m}(2K-1)^{2}=\frac{1}{3}m(4m^{2}-1) \tag{6.3.24}$$

$$\sum|a(k)|^{3}=\sum_{K=1}^{m}(2K-1)^{3}=m^{2}(2m^{2}-1) \tag{6.3.25}$$

$$\sum a^{4}(k)=\sum_{K=1}^{m}(2K-1)^{4}=\frac{1}{15}m(4m^{2}-1)(12m^{2}-7) \tag{6.3.26}$$

对于正方形星座图,序列出现在每个子区域的概率是相同的,即 $1/m$,

所以

$$E[|a(k)|^L]=\frac{1}{m}\sum[|a(k)|^L] \tag{6.3.27}$$

传统的多模盲均衡算法，$L=2$，模值为

$$R_M^2=\frac{E[a^4(k)]}{E[a^2(k)]}=\frac{12m^2-7}{5} \tag{6.3.28}$$

对于十字形星座图，MMA 对每一维有好几个模值。各个模值的大小可用式(6.3.25)及式(6.3.26)求解。

通过式(6.3.28)的计算方法，我们可将 MMA 的代价函数重新写为

$$J_{\text{MMA-min}}(k)=E[(a^2(k)-R_M^2)]=R_M^4-E[a^4(k)] \tag{6.3.29}$$

联立式(6.3.27)～式(6.3.29)，得

$$\begin{aligned}J_{\text{MMA-min}}(k)&=\frac{1}{25}(12m^2-7)^2-\frac{1}{15}(4m^2-1)(12m^2-1)\\&=\frac{16}{76}(12m^4-19m^2+7)\\&=\frac{16}{76}(12m^2-7)(m^2-1)\end{aligned} \tag{6.3.30}$$

## 6.3.2 广义多模盲均衡算法

广义多模盲均衡算法原理(除去虚线框内)如图 6.9 所示。

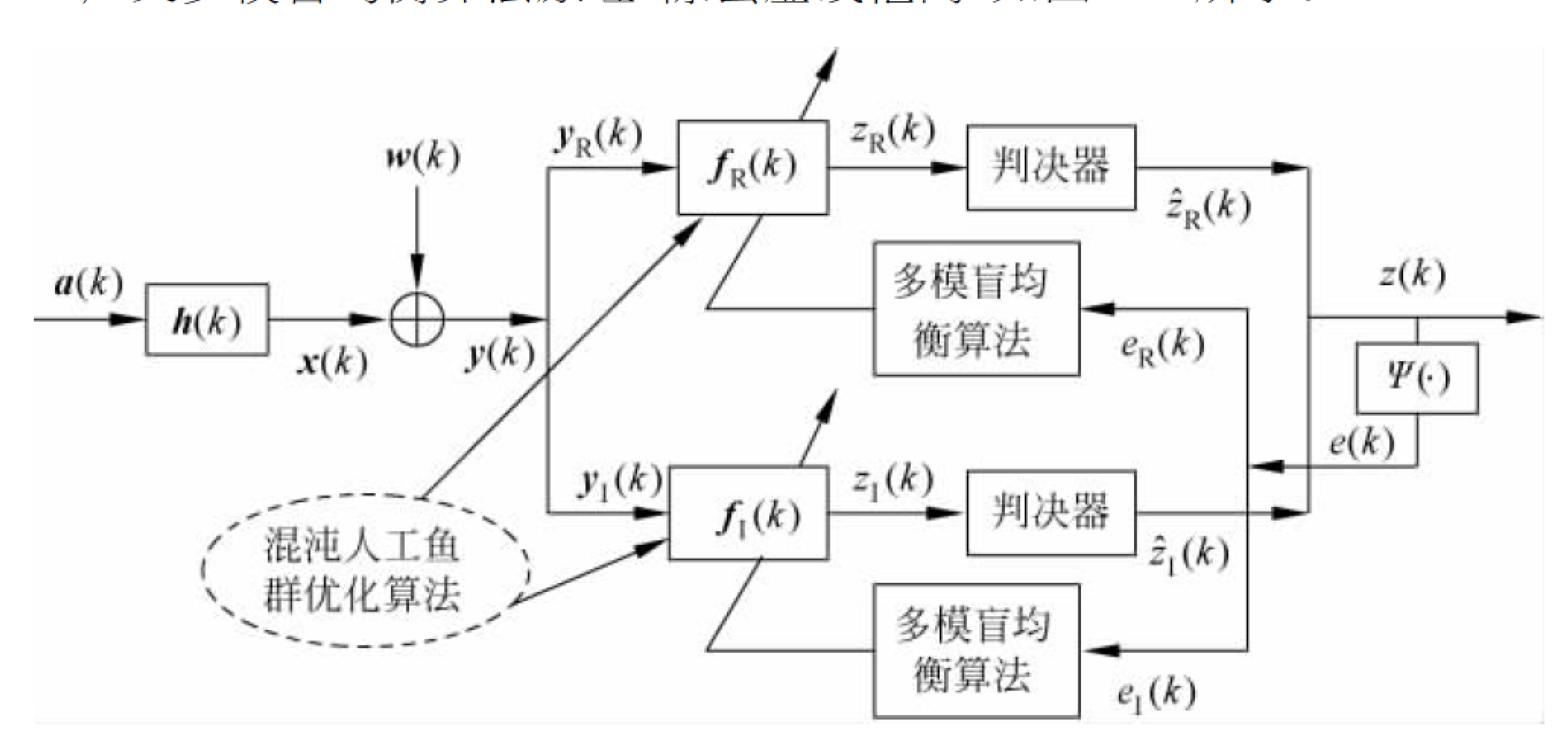



图 6.9 广义多模盲均衡算法原理图

图中，$\boldsymbol{a}(k)$为独立同分布且均值为零的发射信号；$\boldsymbol{h}(k)$是信道脉冲响应向量；$\boldsymbol{w}(k)$为加性高斯白噪声向量；$\boldsymbol{y}(k)=\boldsymbol{y}_R(k)+\mathrm{j}\boldsymbol{y}_\mathrm{I}(k)$为均衡器输入信号；$\boldsymbol{f}(k)$为均衡器权向量；$\Psi(\cdot)$是误差生成函数；$z(k)$是均衡器的输出复信号。

当 MMA 稳态收敛以后，误差修正函数$(z^2(k)-R_\mathrm{M}^2)$的均值趋于零，但其方差即使在无噪声情况下也不为零，导致超量均方误差(MSE)比较大。有两种方法控制稳态误差[13]：①减小迭代步长 $\mu$；②降低误差函数$(z^2(k)-R_\mathrm{M}^2)$。然而，$\mu$ 受收敛速度和精度的制约，不可能无限减小，因此降低误差函数显得更为有效。随着信号调制阶数的增加，误差函数对超量均 MSE 的贡献也越来越大。如果利用星座图的先验信息，选择合适的代价函数与 QAM 星座图匹配，就能进一步减小均方误差[14]。

基于这种思想，将星座图分为几个不同的空间子集，如图 6.10 所示。对处于不同空间子集的输入信号，按照不同的模值对其均衡，获得了广义多模盲均衡算法(GMMA)，其代价函数为

$$J_\mathrm{GMMA}(k)=E[(z_{\mathrm{R},n}^2(k)-R_{\mathrm{R},n}^2)^2+(z_{\mathrm{I},n}^2(k)-R_{\mathrm{I},n}^2)^2],\quad n=1,2,3,\cdots \tag{6.3.31}$$

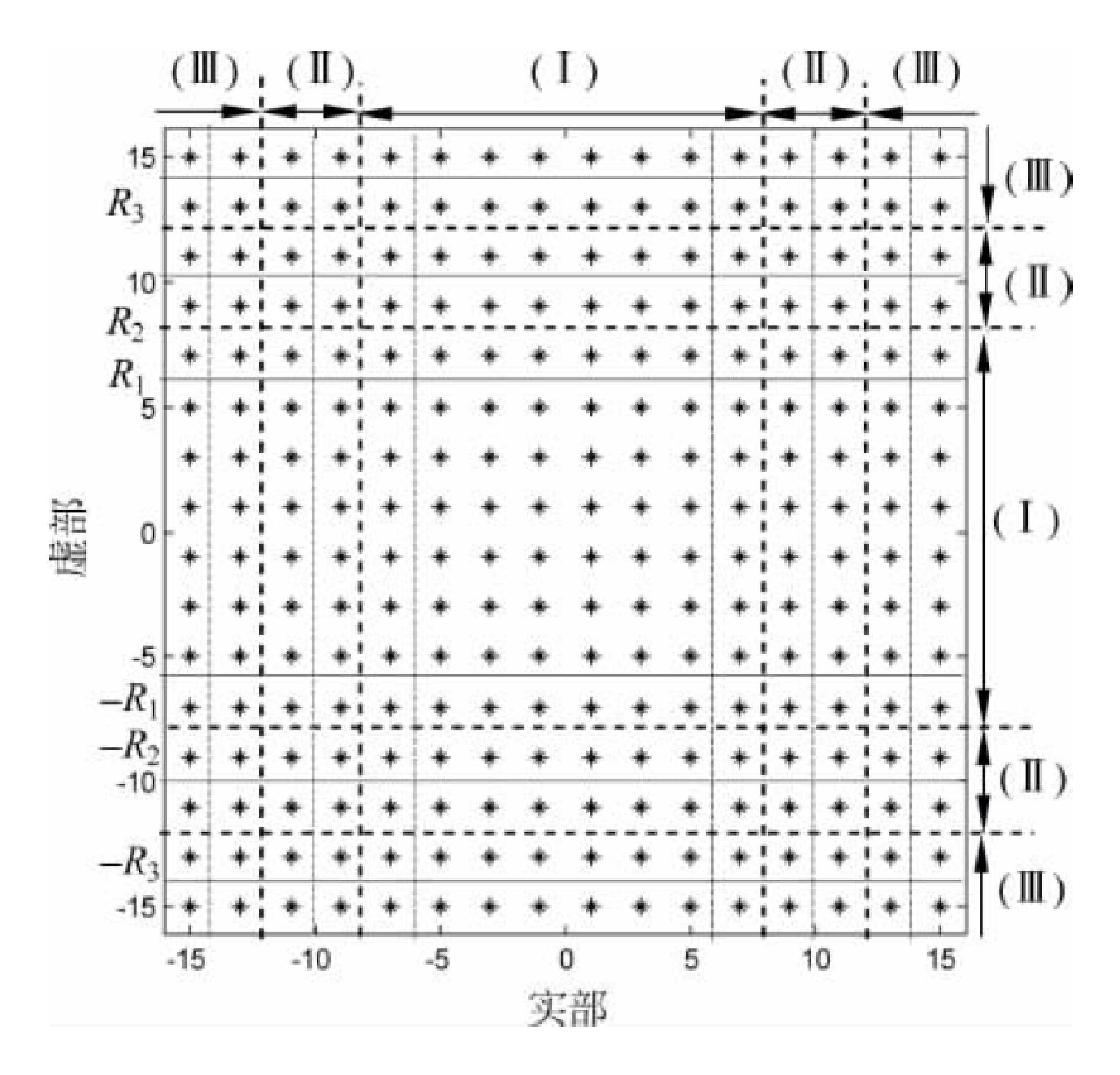



图 6.10 256-QAM 星座图

式中，$z_{\mathrm{R},n}^2(k)$和$z_{\mathrm{I},n}^2(k)$分别为第$k$时刻第$n$个区域的均衡器输出信号的实部与虚部；$R_{\mathrm{R},n}$和$R_{\mathrm{I},n}$分别表示在区域$n$中模值的实部或虚部。下面仅以同相分量来分析，第$n$个区域的输出信号$\{z_{\mathrm{R},n}(k)\}$的广义多模盲均衡算法的模值和代价函数为

$$R_{\mathrm{R},n}^2 = \frac{E[a_{\mathrm{R},n}^4(k)]}{E[a_{\mathrm{R},n}^2(k)]} \tag{6.3.32}$$

$$J_{\mathrm{GMMA\ R},n}(k) = E[(a_{\mathrm{R},n}^2(k) - R_{\mathrm{R},n}^2)^2] = R_{\mathrm{R},n}^4 - E[a_{\mathrm{R},n}^4(k)] \tag{6.3.33}$$

式中，$\{a_{\mathrm{R},n}(k)\}$属于输出信号子集$\{z_{\mathrm{R},n}(k)\}$中的某一采样符号，且

$$E[a_{\mathrm{R},n}^{\lambda}(k)] = \frac{1}{m_n - m_{n-1}}\left[\sum_{k=1}^{m_n}(2k-1)^{\lambda} - \sum_{k=1}^{m_{n-1}}(2k-1)^{\lambda}\right], \quad \lambda = 2,4 \tag{6.3.34}$$

式中，$m_n$和$m_{n-1}$是正方形星座图中第$n$和$n-1$区域中的码元数目。利用式(6.3.24)和式(6.3.26)，得第$n$个区域GMMA的代价函数为

$$J_{\mathrm{GMMA\ R},n}(k) = \frac{1}{25}\frac{(48m_n^5 - 40m_n^3 + 7m_n - 48m_{n-1}^5 - 40m_{n-1}^3 + 7m_{n-1})^2}{(4m_n^3 - m_n - 4m_{n-1}^3 - m_{n-1})^2} - \frac{1}{15}\frac{48m_n^5 - 40m_n^3 + 7m_n - 48m_{n-1}^5 - 40m_{n-1}^3 + 7m_{n-1}}{m_n - m_{n-1}} \tag{6.3.35}$$

当输出信号$z(k)\to a(k)$或$z_{\mathrm{R},n}(k)\to a_{\mathrm{R},n}(k)$时，代价函数式(6.3.31)获最小值。假设输入信号$a_{\mathrm{R}}(k)$或$a_{\mathrm{I}}(k)$的取值在$\{\pm1,\pm3,\cdots,\pm(2m+1)\}$区域时，则信号最大幅度调制为$m$。GMMA迭代步骤(仅考虑同相分量)如下。

步骤1：寻找到最小化GMMA代价函数的一个目标值，并将其星座图分解成几个与之相关的正方形星座图，同时将它们的代价函数联系起来，而每一个相关的正方形星座图的代价函数可通过MMA获得。以图6.10为例，此图为256星座图。在区域(Ⅰ)中($m=4$)，从式(6.3.30)我们可计算出此区域代价函数的最小值为$J_{\mathrm{MMA\text{-}min}}(k)=592$。即：

$$m_1 = 4 \quad 0 \leqslant |z_{\mathrm{R},1}(k)| \leqslant 8 \quad R_1 \approx 6.08$$

步骤2：观察图6.10，发现当输出信号处于区域(Ⅱ)中时，令$m_2>m_1$。

通过新增加的 $m_2-m_1$ 个码元数，则处于区域（Ⅱ）中输入信号在子集 $\{\pm(2m_1+1),\pm(2m_1+3),\cdots,\pm(2m_2-1)\}$ 范围内。式(6.3.35)给出了输出信号处在第 $n$ 区域时代价函数最小值的一般表达式。将此式具体化到区域（Ⅱ），得

$$J_{\mathrm{GMMA,R}}(2)=\frac{1}{25}\frac{(48m_2^5-40m_2^3+7m_n-46\,620)^2}{(4m_2^3-m_n-252)^2}-\frac{1}{15}\frac{48m_2^5-40m_2^3+7m_2-46\,620}{m_2-4} \tag{6.3.36}$$

而此时，问题的关键是寻找 $m_2$ 的值，计算此区域代价函数的最小值而寻找到变量 $m_2$ 的值。

当 $m_2=6$ 时，代价函数的最小值接近目标值 $J_{\mathrm{MMA\text{-}min}}(k)=592$，即

$$m_2=6\quad 8\leqslant|z_{\mathrm{R},2}(k)|\leqslant 12\quad R_2\approx 10.25$$

步骤 3：设计区域（Ⅲ）中的参数，其方法步骤与步骤 2 相同。找到区域（Ⅲ）中的参数：

$$m_3=8\quad 12\leqslant|z_{\mathrm{R},3}(k)|\leqslant\infty\quad R_3\approx 14.17$$

重复迭代以上 3 个步骤直到 $m_n$ 的值大于 $m$。

与 MMA 的代价函数式(6.3.3)相比，GMMA 用 $R_{\mathrm{R},K}$ 和 $R_{\mathrm{I},K}$ 代替了 $R_{\mathrm{M}}$。其中，$R_{\mathrm{M}}$ 是只与信源统计特性相关的常数；而 $R_{\mathrm{R},K}$ 和 $R_{\mathrm{I},K}$ 不仅与信源的统计特性相关，还与判决器输出有关。在迭代过程中，当 $z_{\mathrm{R},K}(k)$ 和 $z_{\mathrm{I},K}(k)$ 处在不同空间子集时，$R_{\mathrm{R},K}$ 和 $R_{\mathrm{I},K}$ 自适应地改变取值。

### 6.3.3 基于混沌人工鱼群算法优化的广义多模盲均衡算法

#### 1. 混沌人工鱼群优化算法[15]

人工鱼群算法（AFSA）是一种基于种群智能搜索策略的全局优化算法。人工鱼群中每条鱼都具有位置和食物浓度两个特征。在寻优过程中，根据人工鱼搜索到的历史最优值和群体中其他人工鱼搜索到的最优值，不断更新当前人工鱼的位置，通过判断人工鱼所在位置的食物浓度来选择最

优值。

对于高维度复杂优化问题，AFSA 存在易陷入局部极值和算法后期寻优效率低等问题。人工鱼群的初始化具有随机性，导致种群中部分人工鱼远离最优解；若初始种群比较好，则能提高寻优效率和解的精度。混沌序列具有遍历性和规律性的特点，混沌优化算法利用这一特征，将人工鱼的位置向量映射为混沌变量，通过迭代产生混沌序列，并作逆映射得到新的位置向量，将新的位置向量作为人工鱼的初始位置向量。本节常用的一维 Logistic 映射，该映射模型为

$$q(k+1) = \rho q(k)[1-q(k)] \tag{6.3.37}$$

式(6.3.37)表明，混沌变量第 $k$ 与 $k+1$ 次的迭代关系，$\rho\in(2,4]$被称为 Logistic 参数。AFSA 在搜索后期，大量人工鱼在局部极值点处聚集导致算法“早熟”。因此，在聚群和追尾行为后利用混沌遍历和随机微扰动使鱼群摆脱局部极值点，提高算法后期的搜索效率，这就是混沌人工鱼群优化算法(Chaotic Artificial Fish Swarm Algorithm，CAFSA)。

### 2. 基于混沌人工鱼群算法优化的广义多模盲均衡算法[15]

本节利用 CAFSA 优化均衡器权向量，其原理如图 6.9 所示。结合广义多模盲均衡算法(GMMA)的代价函数，确定 CAFSA 的适应度函数，通过 CAFSA 寻找人工鱼群的最优位置向量，并作为广义多模盲均衡算法的初始权向量，优化步骤如下。

步骤 1：初始化人工鱼群。随机产生人工鱼的位置和所在位置的食物浓度。

步骤 2：鱼群混沌初始化。将人工鱼的位置向量由 Logistic 映射，得到新的位置向量。

步骤 3：计算每条人工鱼的适应度值。通过觅食、聚群和追尾行为来更新每条人工鱼的位置向量和浓度，最优值写入公告板。适应度函数定义为

$$\mathrm{Fit}(\boldsymbol{X}_i) = 1/J(\boldsymbol{X}_i) \tag{6.3.38}$$

式中，$J(\boldsymbol{X}_i)=J_{\text{CAFSA-GMMA}}(\boldsymbol{f})$为均衡器的代价函数，$\boldsymbol{f}$ 是 CAFSA-GMMA 的权向量，$\boldsymbol{X}_i$ 是人工鱼的位置向量。

步骤 4：计算人工鱼的适应度方差 $\sigma^2$。如果 $\sigma^2<\varepsilon$，说明算法早熟，进行混沌微扰动转到步骤 5，否则转步骤 6。适应度方差定义为

$$\sigma^2=\sum_{i=1}^{D}\left(\frac{\text{Fit}(\boldsymbol{X}_i)-\text{Fit}_{\text{avg}}}{\text{Fit}}\right)^2 \tag{6.3.39}$$

式中，$D$ 为人工鱼的维数；$\text{Fit}(\boldsymbol{X}_i)$为第 $i$ 条人工鱼的适应度值；$\text{Fit}_{\text{avg}}$为当前人工鱼群的平均适应度值；Fit 为归一化因子，其作用是限制 $\sigma^2$ 的大小。Fit 的计算公式为

$$\text{Fit}=\begin{cases}\max\limits_{1\leqslant i\leqslant D}|\text{Fit}_i-\text{Fit}_{\text{avg}}|, & \max|\text{Fit}_i-\text{Fit}_{\text{avg}}|>1\\ 1, & \text{其他}\end{cases} \tag{6.3.40}$$

步骤 5：混沌扰动。将人工鱼的位置向量在视野范围[－Visual，Visual]内混沌搜索

$$\boldsymbol{X}_{i|\text{next}}=\boldsymbol{X}_i+\Delta_i\boldsymbol{e}_i \tag{6.3.41}$$

得到新的位置向量后转到步骤 6。式中

$$\Delta_i=-\text{Visual}+\rho*\text{Visual}*q_d \tag{6.3.42}$$

$q_d$ 为经过混沌迭代后的混沌变量；$\boldsymbol{e}_i$ 为单位向量。

步骤 6：行为选择。通过 CAFSA 优化得到新的种群，将新种群中的最优值与公告板比较，若优于公告板，则更新公告板。

步骤 7：判断是否达到最大进化次数。若达到，退出循环；否则，跳转步骤 3。

因此，上述步骤 1～步骤 7 构成混沌人工鱼群优化算法。其中，人工鱼群算法解决的是宏观全局寻优问题；而混沌算法处理的是微观局部加速问题。将两者融合使用，可以充分发挥各自的优势，获得更好的优化效果。

### 6.3.4 算法仿真

为了验证 CAFSA-GMMA 的性能，以 GMMA 和 AFSA-GMMA 为比

较对象，进行仿真实验。

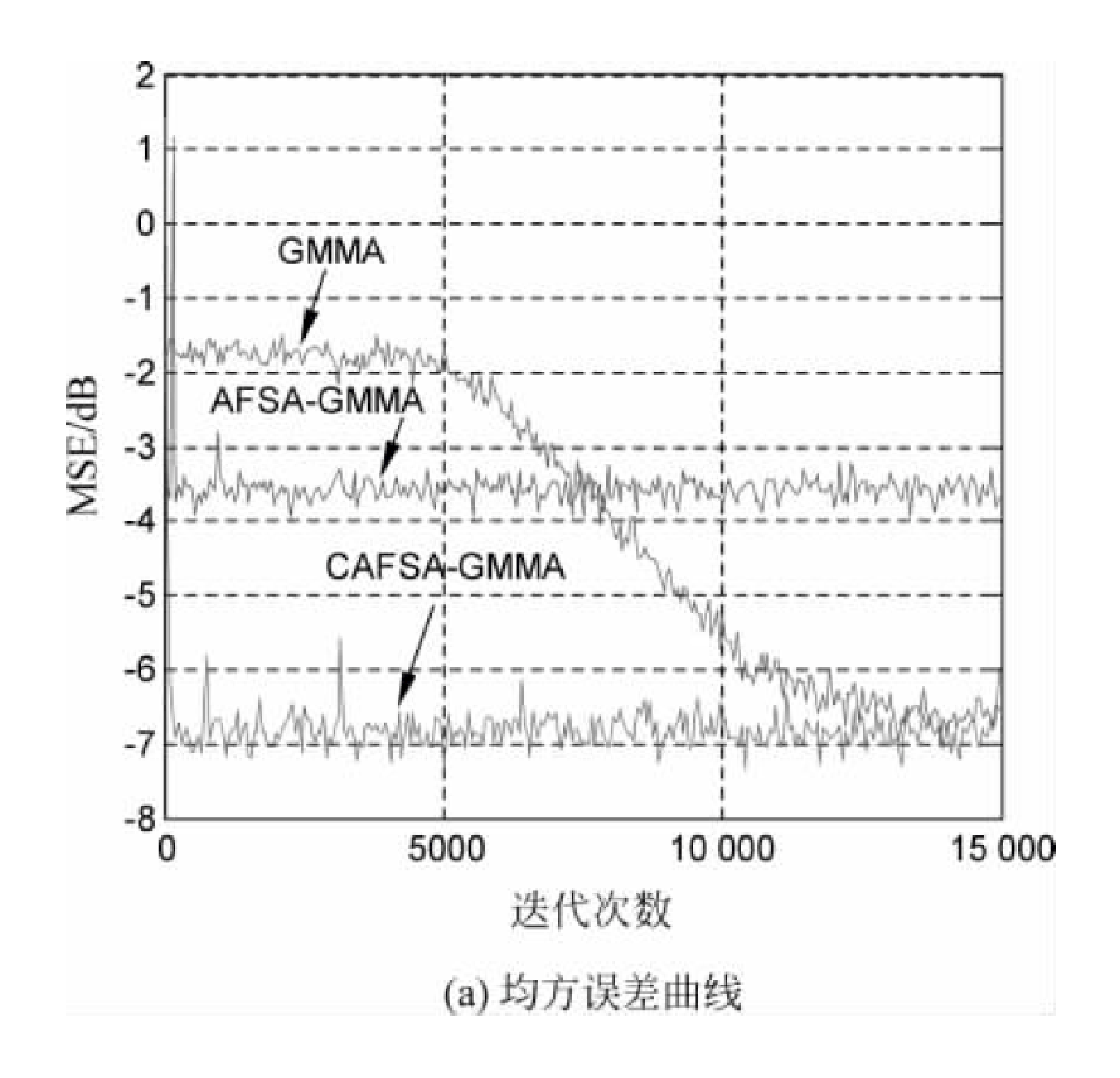
**【实验 6.2】** 最大相位信道 $\boldsymbol{h}=[-0.35\quad 0\quad 0\quad 1]$，发射信号为 256QAM，信噪比为 30dB。均衡器权长为 8，GMMA、AFSA-GMMA 和 CAFSA-GMMA 的模值，如表 6.3 所示。

表 6.3 不同区域模值的大小

| $R_1$ | $R_2$ | $R_3$ |
|---|---|---|
| 6 | 10.35 | 14.2 |

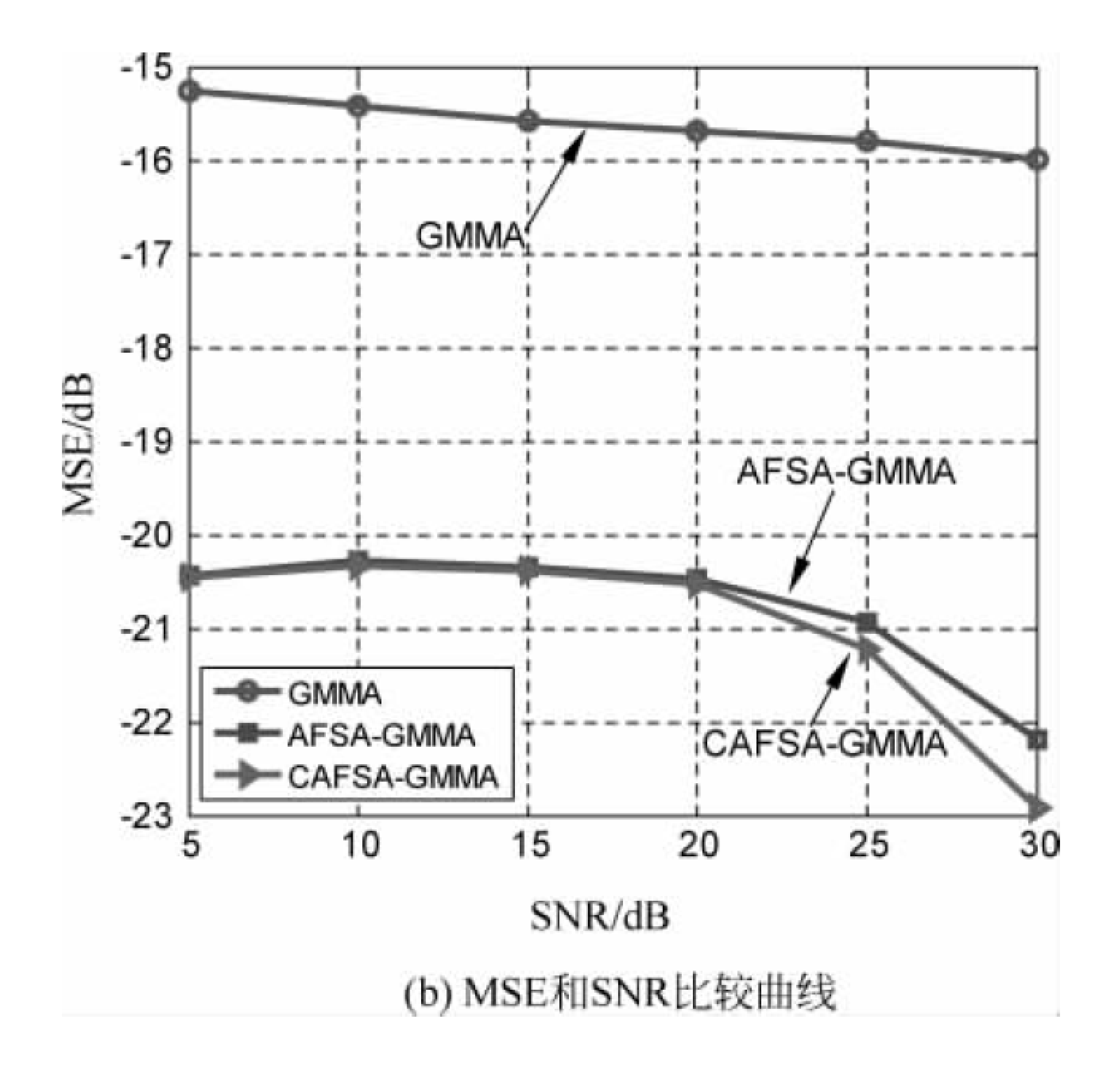
GMMA 采用中心抽头初始化，步长为 $2\times10^{-6}$，种群规模为 50，进化次数为 20；AFSA-GMMA 和 CAFSA-GMMA 步长分别为 $1.8\times10^{-6}$ 和 $2.5\times10^{-6}$，视野范围 0.5。600 次蒙特卡罗仿真结果如图 6.11 所示。

图 6.11 表明，CAFSA-GMMA 的收敛速度比 GMMA 快约 12 000 步，与 AFSA-GMMA 基本一致；在稳态误差上，与 GMMA 基本相同，与 AFSA-GMMA 相比，减少了 3dB；而且，CAFSA-GMMA 的输出星座图更加清晰、紧凑。



(a) 均方误差曲线

图 6.11 仿真结果




(b) MSE和SNR比较曲线

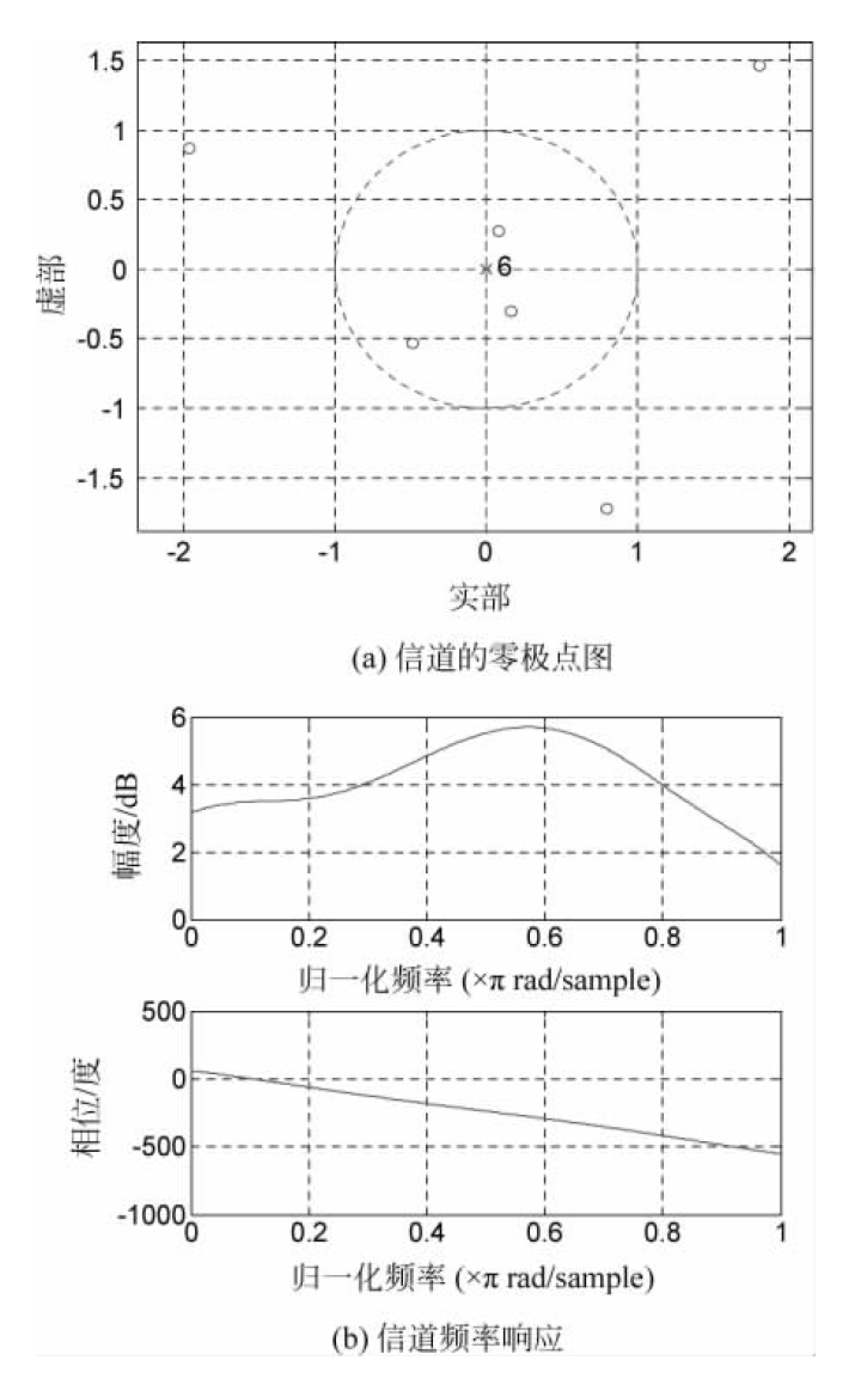
(c) GMMA输出信号

(d) AFSA-GMMA输出信号

(e) CAFSA-GMMA输出信号

图 6.11 （续）

**【实验 6.3】** 采用音频带宽信道，其零点与频率分布，如图 6.12 所示。GMMA，AFSA-GMMA 和 CAFSA-GMMA 步长分别为 $2\times10^{-6}$，$1.8\times10^{-6}$，$1\times10^{-6}$，其他参数与实验 6.2 一致。600 次蒙特卡罗仿真结果如图 6.13 所示。

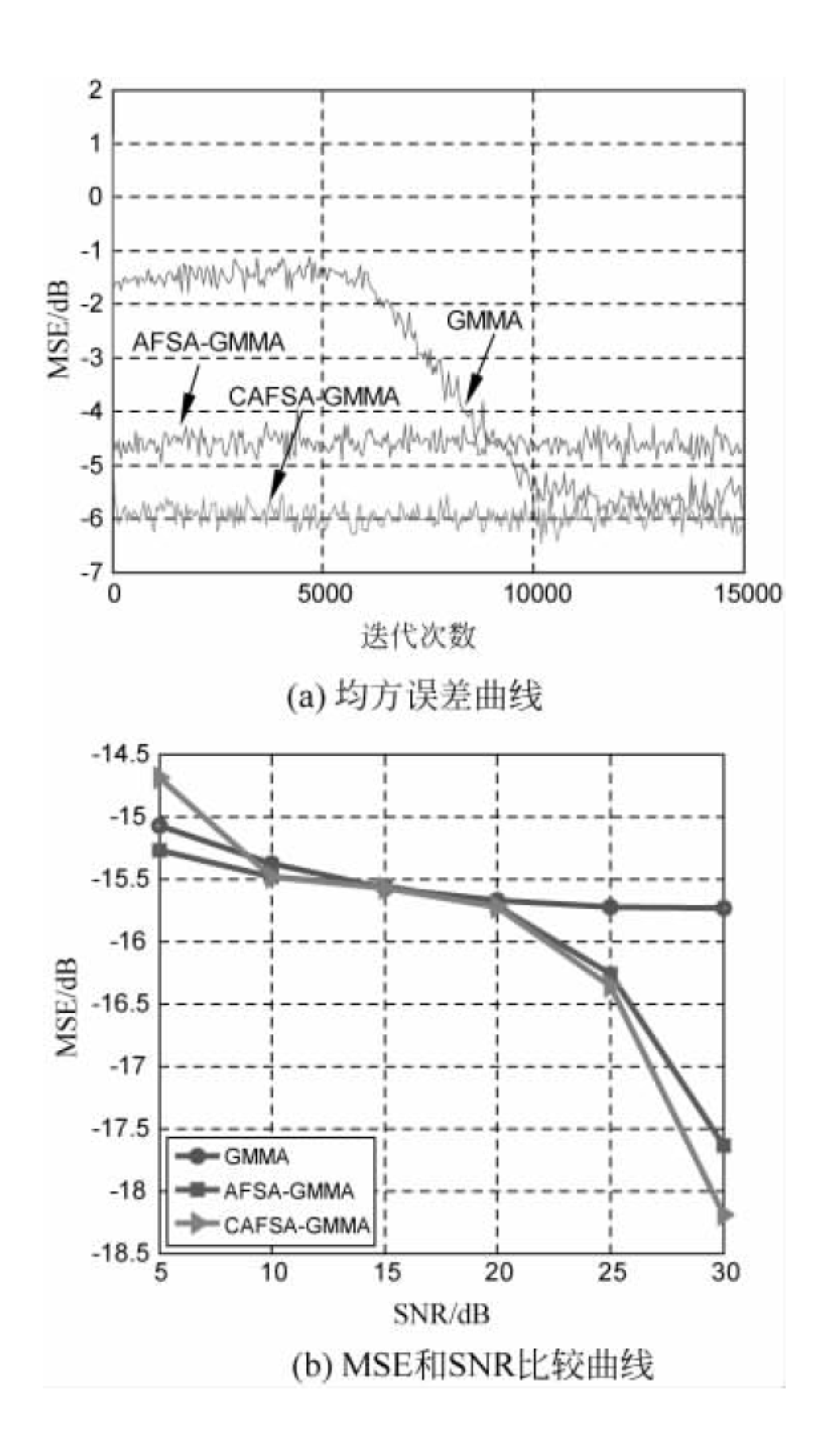


(a) 信道的零极点图

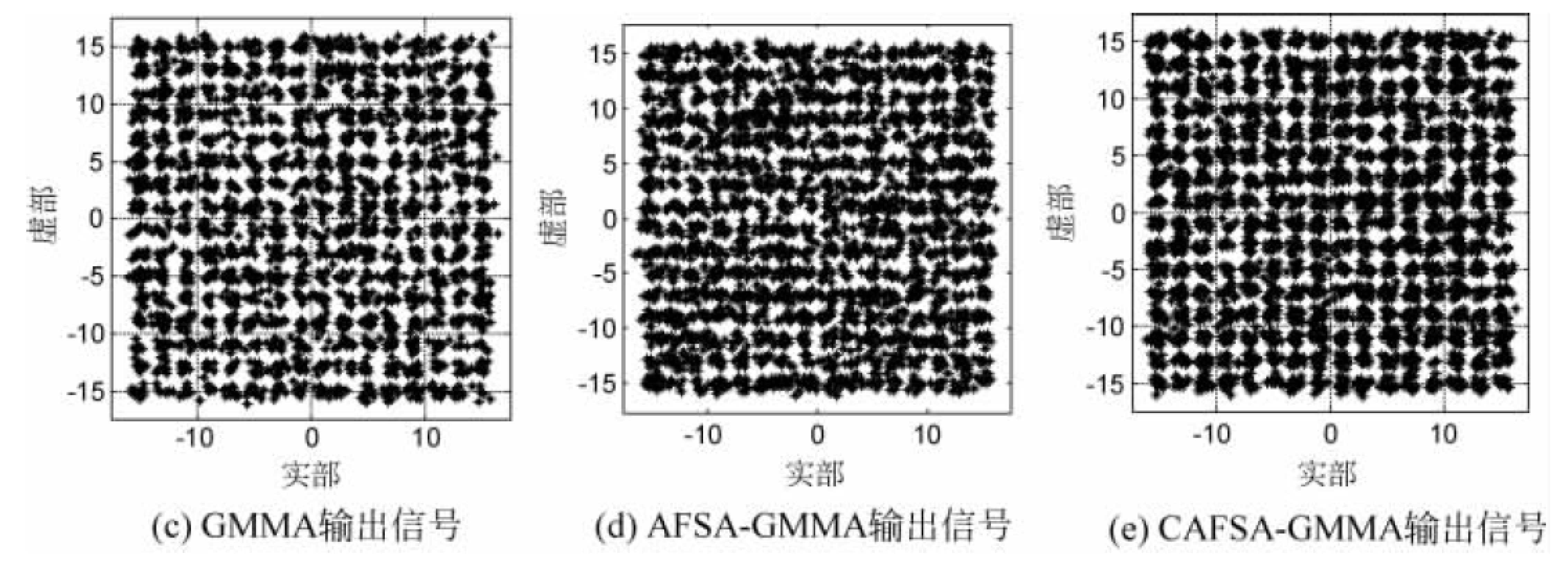
(b) 信道频率响应

图 6.12 信道特性

图 6.13 表明，CAFSA-GMMA 的收敛速度比 GMMA 快约 11 000 步，与 AFSA-GMMA 基本一致；在稳态误差上，与 GMMA 基本相同，与 AFSA-GMMA 相比，减少了 1dB。而且，CAFSA-GMMA 的输出星座图更加清晰，紧凑。



(a) 均方误差曲线
(b) MSE和SNR比较曲线



(c) GMMA输出信号　(d) AFSA-GMMA输出信号　(e) CAFSA-GMMA输出信号

图 6.13　仿真结果

## 6.4 基于免疫人工鱼群算法优化的正交小波频域多模盲均衡算法

自适应多模盲均衡算法均衡更高阶 QAM 信号时，提高收敛速度和降低稳态误差方面效果不是很理想。为此，本节在自适应多模盲均衡算法基础上，将正交小波变换、快速傅里叶变换(FFT)技术和免疫人工鱼群算法有机融合，研究了基于免疫人工鱼群算法优化的正交小波频域自适应多模盲均衡算法（Immune Artificial Fish Swarm Algorithm based Frequency domain orthogonal Wavelet Transform βMMA，ICAFSA-FWT-βMMA）。该算法利用免疫人工鱼群算法快速寻优能力来加快自适应多模盲均衡算法的收敛速度；利用正交小波自适应变换对输入信号良好的去相关性来减小稳态误差，并用循环卷积代替线性卷积减小计算量。与自适应多模盲均衡算法（βMMA）、正交小波多模盲均衡算法（orthogonal Wavelet Transform βMMA，WT-βMMA）和正交小波频域自适应多模盲均衡算法(Frequency domain orthogonal Wavelet Transform βMMA，FWT-βMMA)相比，该算法具有更小的均方误差和更快的收敛速度。

### 6.4.1 频域盲均衡算法

频域 LMS 算法[16]的基本思想是将均衡器时域输入信号分成长度为 $L$ 的块($L$ 为均衡器的权长)，权向量每 $L$ 样点进行一次更新，每次更新由 $L$ 个误差信号样点累加结果进行控制，这就保证了与时域 LMS 算法有相同的收敛速度。同时利用快速傅里叶变换、时域输入信号重叠保留法(overlap-save)和频域乘积代替时域信号的线性卷积，实现频域权向量更新，大大降低计算量。而且，如果时域输入信号具有很强的自相关性，通过快速傅里叶变换变换到频域后，使这种相关性被分散开，有利于加快收敛速度。

频域盲均衡算法将频域 LMS 算法的误差信号定义为不需要期望响应

信号的“盲”形式，不妨设频域误差函数为

$$E(k) = R_{\mathrm{F}}^2 - |Z(k)|^2 \tag{6.4.1}$$

由 CMA 时域权向量更新公式，得频域盲均衡算法的权向量更新公式为

$$\boldsymbol{F}(k+1) = \boldsymbol{F}(k) + \mu E(k) Z(k) \boldsymbol{Y}^*(k) \tag{6.4.2}$$

式中，$R_{\mathrm{F}}^2$ 为信号统计模值 $R^2 = E[|a(k)|^4]/E[|a(k)|^2]$的快速傅里叶变换；$\boldsymbol{a}(k)$为信源发射序列；$Z(k)$为频域盲均衡器的输出信号；$\boldsymbol{F}(k)$表示频域均衡器权向量；$\boldsymbol{Y}(k)$为频域均衡器输入信号向量；$\mu$ 为迭代步长。频域盲均衡算法流程如图 6.14 所示。

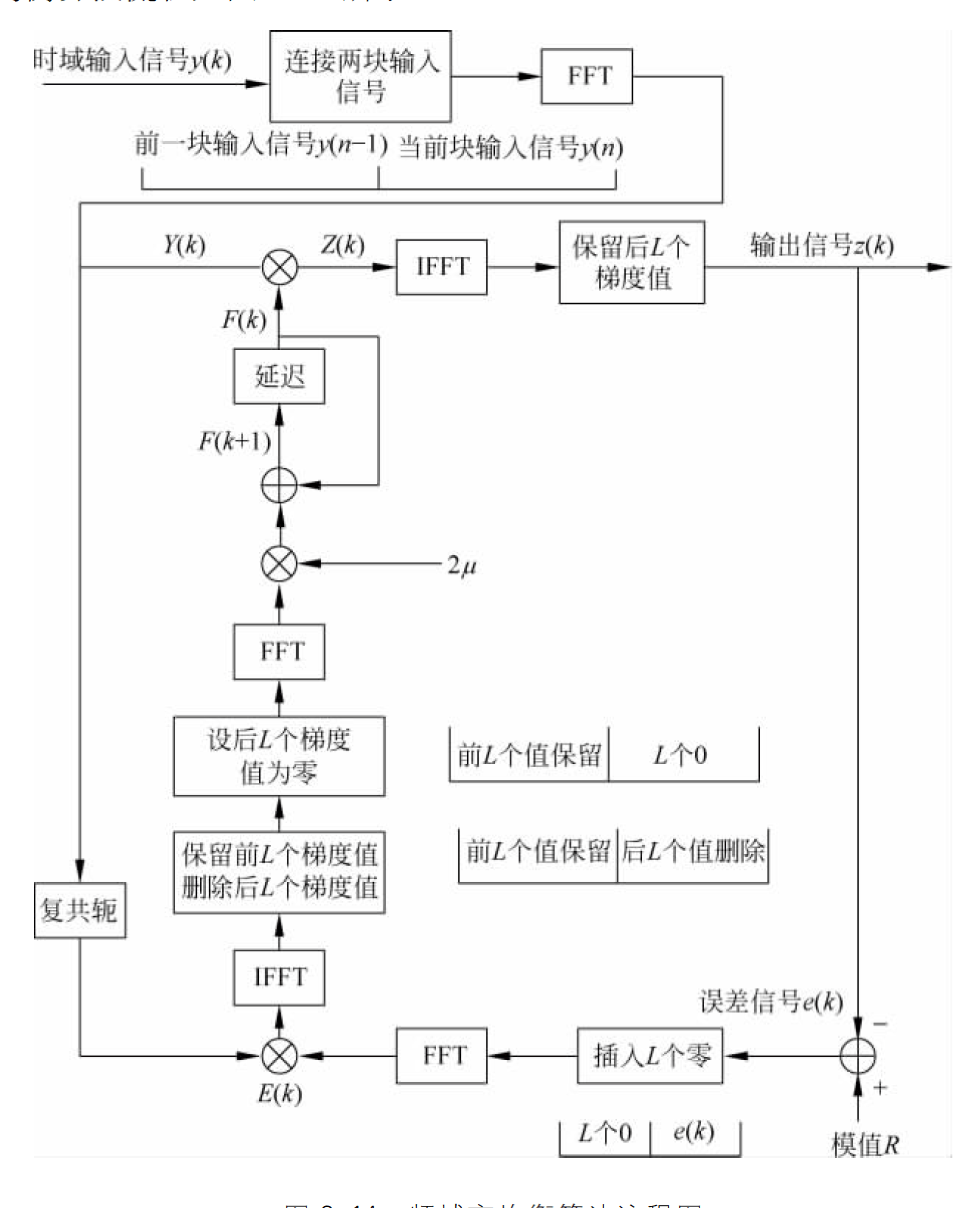



图 6.14 频域盲均衡算法流程图

### 6.4.2　自适应频域正交小波自适应多模盲均衡算法

#### 1. 自适应时域多模盲均衡算法

文献[17]提出了一种非均方误差代价函数为

$$J = \max_f E[|z(k)|^2] \tag{6.4.3}$$

式(6.4.3)的约束条件为

$$f_{\max}(\gamma, |z_R(k)|) = \gamma \tag{6.4.4}$$

$$f_{\max}(\gamma, |z_I(k)|) = \gamma \tag{6.4.5}$$

式中，$z(k)$为均衡器的输出信号；$\gamma$为原始序列$\boldsymbol{a}(k)$的最大幅度；$\boldsymbol{f}$为均衡器权向量；对任意的实数$a,b>0$，$f_{\max}$定义为

$$f_{\max}(a,b) = \frac{|a+b|+|a-b|}{2} = \begin{cases} a, & a \geqslant b \\ b, & b \geqslant a \end{cases} \tag{6.4.6}$$

函数$f_{\max}$是可微的，对任意的$a,b \in R$，有

$$\frac{\partial}{\partial a} f_{\max}(|a|,|b|) = \frac{\text{sgn}[a]}{2}(1+\text{sgn}[|a|-|b|]) \tag{6.4.7}$$

对于星座图为正方形的MQAM信号，若均衡器输出$z(k)$落在边长为$\gamma$的正方形区域$\Re$内，则代价函数式(6.4.3)的约束条件式(6.4.4)成立。输出信号的平均能量$E[|z(k)|^2]$接近原始信号的平均能量$E[|a(k)|^2]$。因此，利用拉格朗日乘数法，将代价函数式(6.4.3)和约束条件式(6.4.4)转化为

$$J(k) = |z(k)|^2 + \lambda_R(f_{\max}(\gamma, |z_R(k)|) - \gamma) + \lambda_I(f_{\max}(\gamma, |z_I(k)|) - \gamma) \tag{6.4.8}$$

对式(6.4.4)求导，则权向量更新公式为

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + \frac{\mu}{4}[\lambda_R g_R + \mathrm{j}\lambda_I g_I + 4z(k)]^* \boldsymbol{y}(k) \tag{6.4.9a}$$

$$g_L = \text{sgn}[z_L(k)](1+\text{sgn}(|z_L(k)|) - \gamma) \tag{6.4.9b}$$

式中，$L$表示$R$或$I$；$\text{sgn}(\cdot)$为符号函数；$\boldsymbol{y}(k)$为均衡器输入信号。

若$|z_L(k)|<\gamma$，则$g_L=0$，约束条件式(6.4.4)和式(6.4.5)满足；若

$|z_L(k)|>\gamma$，则 $g_L=2\text{sgn}[z_L(k)]$。利用 Bassgang 过程，计算出 $\lambda_L$ 的值。

$$\underbrace{E[(0.5\lambda_L \text{sgn}[z_L(k)]+z_L(k))z_L(k-n)]}_{|z_L(k)|>\gamma}+\underbrace{E[z_L(k)z_L(k-n)]}_{|Z_L(k)|<\gamma}=0 \mid \forall n\in Z \tag{6.4.10}$$

当利用权向量更新公式(6.4.9a)使眼图能顺利张开时，式(6.4.10)可以被简化。我们可以得到这样的结论：输出信号 $z(k)$是通过信道后的原序列 $a(k)$和加性噪声的卷积和。当 $n\neq0$ 时，对于独立同分布的发射信号，式(6.4.10)满足：当 $n=0$ 时，如果约束条件满足，则需假设：

$$\lambda_L=-2(1+\beta)\mid z_L(k)\mid,\quad \beta>0 \tag{6.4.11}$$

$$\beta=\frac{M-2\sqrt{M}+3}{3\sqrt{M}-3} \tag{6.4.12}$$

联立式(6.4.8)～式(6.4.12)，则自适应时域多模盲均衡算法(βMMA)的代价函数和权向量更新公式为

$$J(k)=E[\mid\lambda_R\mid\mid z_R^2(k)-\gamma^2\mid]+E[\mid\lambda_I\mid\mid z_I^2(k)-\gamma^2\mid] \tag{6.4.13}$$

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu(b_R z_R(k)+b_I z_I(k))^*\boldsymbol{y}(k) \tag{6.4.14}$$

$$b_L=\begin{cases}1, & \mid z_L(k)\mid\leqslant\gamma\\ -\beta, & \mid z_L(k)\mid>\gamma\end{cases} \tag{6.4.15}$$

2. 自适应频域正交小波多模盲均衡算法

自适应频域正交小波自适应多模盲均衡算法原理如图 6.15 所示。

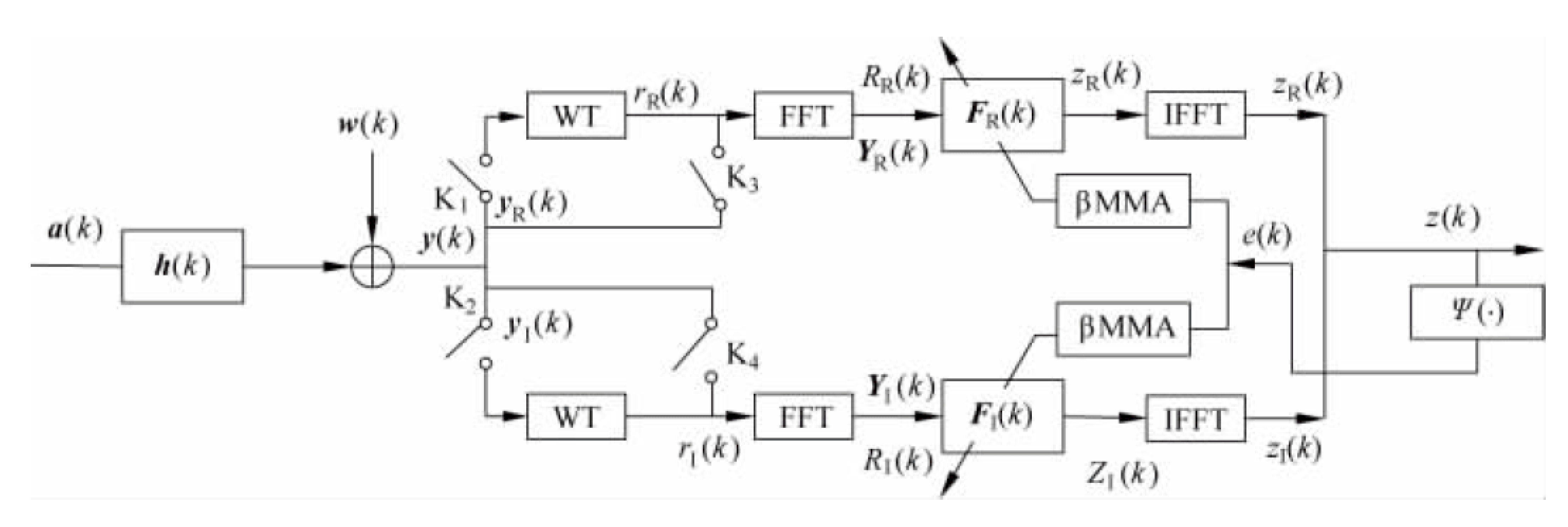



图 6.15　自适应频域小波多模盲均衡算法原理图

图中，当开关 $K_1$、$K_2$ 断开，$K_3$、$K_4$ 闭合时，为自适应频域多模盲均衡算法，即 F-βMMA。$\boldsymbol{h}(k)$表示信道的脉冲响应向量；$\boldsymbol{w}(k)$是加性高斯白噪声向量；$\boldsymbol{y}(k)=[y(k),y(k-1),\cdots,y(k-M_f+1)]^{\mathrm{T}}$ 是均衡器输入信号(上标 T 表示转置)向量；$\boldsymbol{F}(k)=[F(k),F(k-1),\cdots,F(k-M_f+1)]$是频域均衡器权向量(其中 $M_f$ 为均衡器长度)；$Z_{\mathrm{R}}(k)$和 $Z_{\mathrm{I}}(k)$分别表示频域盲均衡器输出信号的实部与虚部。均衡器输入信号在通过均衡器前先将其拆成实部与虚部并分别进行快速傅里叶变换，将时域信号变换到频域。在频域均衡器中，输入信号 $\boldsymbol{y}_{\mathrm{R}}(k)$和 $\boldsymbol{y}_{\mathrm{I}}(k)$分成长为 $M_f$ 的数据块。为避免数据块间的码间干扰，在各个数据块前加上循环保护前缀，使数据块的长度不小于信道阶数，即 $\boldsymbol{Y}_{\mathrm{L}}(k)=[Y_{\mathrm{L}}(k),Y_{\mathrm{L}}(k-1),\cdots,Y_{\mathrm{L}}(k-M_f+1)]$。具有循环特性的数据块 $\boldsymbol{Y}_{\mathrm{L}}(k)$不仅能消除数据块间的码间干扰，还能利用循环卷积的形式将时域卷积等效为频域乘积，大大减少了运算量。

由以上分析，则自适应频域多模盲均衡算法的权向量更新公式为

$$\boldsymbol{F}(k+1)=\boldsymbol{F}(k)+\mu(B_{\mathrm{R}}Z_{\mathrm{R}}(k)+B_{\mathrm{I}}Z_{\mathrm{I}}(k))\boldsymbol{Y}(k) \tag{6.4.16}$$

$$B_{\mathrm{L}}=\begin{cases}1, & |Z_{\mathrm{L}}(k)|\leqslant\gamma\\ -\beta, & |Z_{\mathrm{L}}(k)|>\gamma\end{cases} \tag{6.4.17}$$

3. 自适应频域正交小波多模盲均衡算法

图 6.15 中，当开关 $K_1$、$K_2$ 闭合，$K_3$、$K_4$ 断开时，得到自适应频域正交小波多模盲均衡算法(FWT-βMMA)。该算法将正交小波变换引入到自适应频域多模盲均衡算法中，利用正交小波变换对输入信号进行能量归一化处理，降低输入信号的自相关性，设计出了具有优良性能的均衡器。

由小波分析理论知，经正交小波变换后的均衡器输入信号为

$$\boldsymbol{r}_{\mathrm{L}}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{L}}(k) \tag{6.4.18}$$

式中，$\boldsymbol{V}$ 为小波变换矩阵，均衡器输出为

$$Z_{\mathrm{L}}(k)=\boldsymbol{F}_{\mathrm{L}}(k)\cdot\boldsymbol{R}_{\mathrm{L}}(k)=\boldsymbol{F}_{\mathrm{L}}(k)\cdot\mathrm{FFT}[\boldsymbol{r}_{\mathrm{L}}(k)] \tag{6.4.19}$$

自适应频域正交小波多模盲均衡算法的权向量更新公式为

$$\boldsymbol{F}_{\mathrm{L}}(k+1)=\boldsymbol{F}_{\mathrm{L}}(k)+\mu\boldsymbol{R}_{\mathrm{L}}^{-1}(k)(B_{\mathrm{R}}Z_{\mathrm{L}}(k)+\mathrm{j}B_{\mathrm{I}}Z_{\mathrm{L}}(k))\boldsymbol{R}_{\mathrm{L}}^{*}(k) \tag{6.4.20}$$

式中，$\boldsymbol{R}_{\mathrm{L}}^{*}(k)$为$\boldsymbol{R}_{\mathrm{L}}(k)$的共轭。

$$\boldsymbol{R}_{\mathrm{L}}^{-1}(k)=\mathrm{diag}[\sigma_{j,0}^{2}(k),\sigma_{j,1}^{2}(k),\cdots,\sigma_{j,M_j-1}^{2}(k),\sigma_{j+1,0}^{2}(k),\cdots,\sigma_{j+1,M_{j-1}}^{2}(k)] \tag{6.4.21}$$

式中，$\sigma_{j,n}^{2}(k)$，$\sigma_{j+1,n}^{2}(k)$分别表示小波变换系数$u_{j,n}(k)$，尺度变换系数$s_{j,n}(k)$的平均功率估计，其更新公式为

$$\sigma_{j,n}^{2}(k+1)=\alpha\sigma_{j,n}^{2}(k)+(1-\alpha)\mid u_{jn}^{2}(k)\mid^{2} \tag{6.4.22}$$

$$\sigma_{j+1,n}^{2}(k+1)=\alpha\sigma_{j+1,n}^{2}(k)+(1-\alpha)\mid s_{jn}(k)\mid^{2} \tag{6.4.23}$$

式中，$\alpha$是平滑因子，且$0<\alpha<1$，一般情况下$\alpha$取值接近于1。

### 6.4.3 基于免疫人工鱼群优化的自适应频域正交小波多模盲均衡算法

人工鱼群算法是将“动物逻辑”和“感知反馈”相结合，模拟鱼群的觅食、聚群、追尾和随机等行为的群集智能算法，具有并行搜索方式和对解空间进行快速寻优能力，现已成功运用于大量优化问题的求解中[18-19]。AFSA 的数学模型描述如下。

在一个$D$维目标空间中，种群数目为$N$，其中$\boldsymbol{X}_i=[X_{i1},X_{i2},\cdots,X_{iD}]$表示第$i$条人工鱼的状态（位置）；人工鱼当前位置的食物浓度$Y=\mathrm{Fit}(\boldsymbol{X})$；人工鱼间的距离$d_{i,j}=\|\boldsymbol{X}_i-\boldsymbol{X}_j\|$；人工鱼的感知范围、步长、拥挤度因子和觅食行为时的尝试次数分别为：Visual、Step、$\delta$和 try-number。人工鱼的觅食、聚群和追尾行为的迭代公式为

$$\boldsymbol{X}_j=\begin{cases}\boldsymbol{X}_i+\mathrm{Random(step)}(\boldsymbol{X}_j-\boldsymbol{X}_i)/d_{i,j}, & Y_j<Y_i\\ \boldsymbol{X}_i+\mathrm{Random(step)}, & \text{其他}\end{cases} \tag{6.4.24}$$

$$\boldsymbol{X}_{\mathrm{inext}}=\boldsymbol{X}_i+\mathrm{Random(step)}(\boldsymbol{X}_c-\boldsymbol{X}_i)/d_{i,j} \tag{6.4.25}$$

$$\boldsymbol{X}_{\mathrm{inext}}=\boldsymbol{X}_i+\mathrm{Random(step)}(\boldsymbol{X}_{\max}-\boldsymbol{X}_i)/d_{i,j} \tag{6.4.26}$$

在人工鱼群算法后期，由于人工鱼的同一化作用，算法易陷入局部极

值。免疫克隆选择算法(immune clone algorithm,ICA)将克隆选择原理融入人工免疫算法中,模拟免疫系统功能的一种进化智能算法。该算法中抗体的演化过程主要是算子的克隆选择、变异和增殖,局部和全局搜索能力强,然而收敛速度较慢。因此,利用人工鱼群算法的快速搜索和免疫克隆算法的良好的全局收敛性能,获得的免疫人工鱼群算法(immune clone artificial fish swarm algorithm,ICAFSA)具有快速搜索能力,保证了全局收敛性。将免疫人工鱼群算法引入至自适应频域正交小波多模盲均衡算法中,得到的基于免疫人工鱼群算法优化的自适应频域正交小波多模盲均衡算法(ICAFSA-FWT-βMMA),原理如图 6.16 所示。

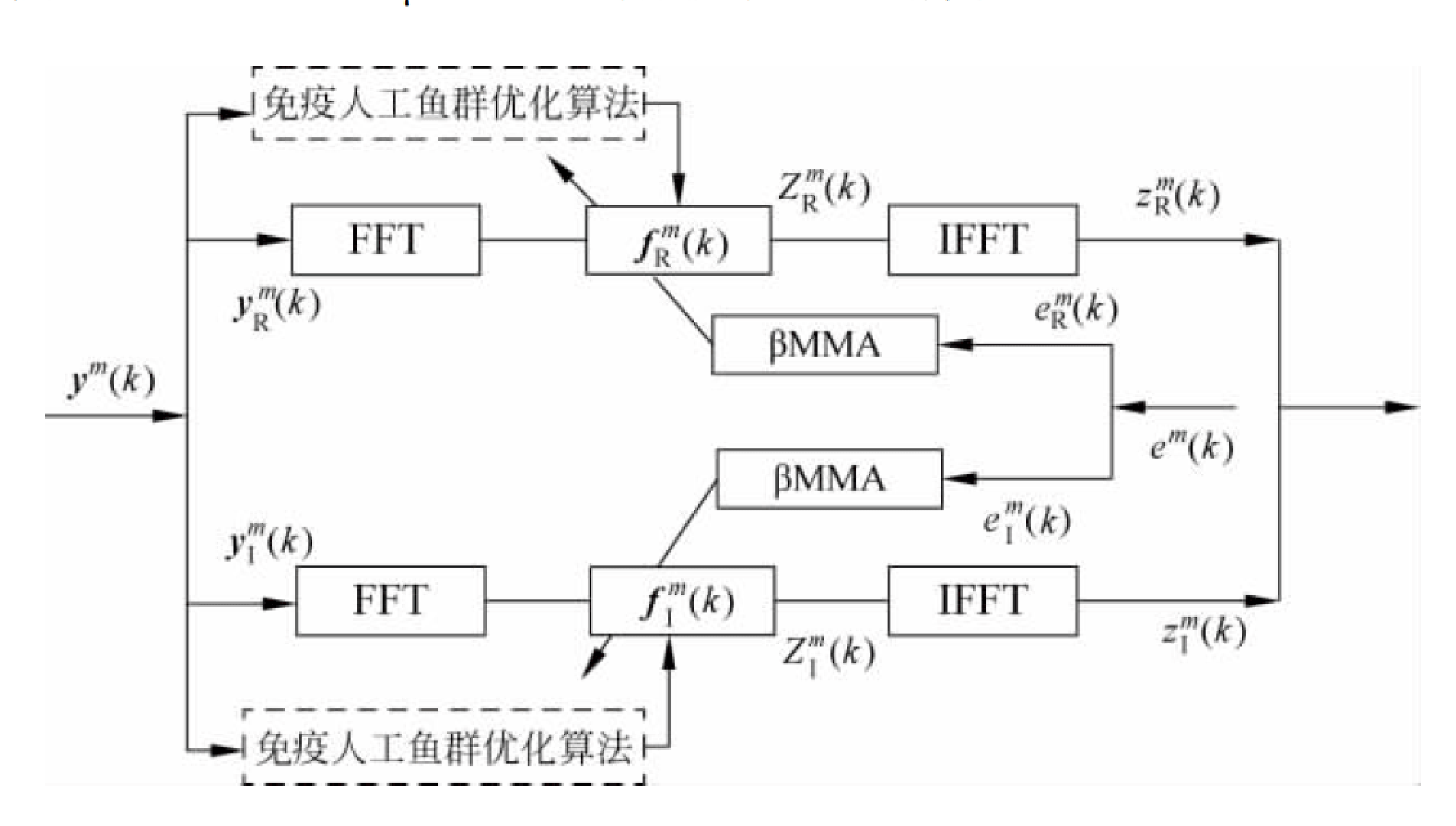



图 6.16 ICAFSA-FWT-βMMA 原理

## 1. 算法流程

将人工鱼群和免疫克隆算法并联,其算法流程如下。

步骤 1:初始化。确定种群规模 $N$,随机产生 $N$ 个个体。

步骤 2:设置公告板。将种群 $N$ 分成 2 个个体数相等的种群,分别为 pop1 和 pop2;前者按人工鱼群算法的适应度函数,找到最优值 $t$;后者按免疫克隆算法抗体的亲和度(适应度),得到最好值 $s$,比较 $s$ 与 $t$,把其中的最优值赋给公告板。

步骤 3:人工鱼群操作。pop1 按人工鱼群算法,进行觅食、聚群、追尾

和随机行为选择，得到最优人工鱼的位置向量 $\boldsymbol{X}_{p\text{best}}$ 和新种群 pop1。

步骤 4：免疫克隆操作。pop2 按免疫克隆算法进行选择、克隆、高频变异和再选择操作，得到最优抗体 $\boldsymbol{X}_{g\text{best}}$ 和新种群 pop2。

步骤 5：更新公告板。比较 $\boldsymbol{X}_{p\text{best}}$ 和 $\boldsymbol{X}_{g\text{best}}$ 的适应度值并与公告板比较，若优于公告板，则更新公告板，否则不变。

步骤 6：终止条件判断。判断最优解是否达到满意的误差范围，若不满足，转到步骤 3，否则转到步骤 7。

步骤 7：算法终止。输出公告板的最优解。

#### 2. 确定适应度函数

盲均衡算法使代价函数迭代至最小值，得到最优的均衡器权向量。而免疫人工鱼群算法在搜索空间中找到适应度值最大的个体。为解决这对矛盾，将自适应频域正交小波多模盲均衡算法代价函数的倒数作为免疫人工鱼群算法的适应度函数，则

$$\text{Fit}(\boldsymbol{f}_i) = 1/J_{\text{FWT-βMMA}}(\boldsymbol{f}_i), \quad i = 1,2,\cdots,N \tag{6.4.27}$$

式中，$J(f_i)=J_{\text{FWT-βMMA}}$ 是均衡器的代价函数，$\boldsymbol{f}_i$ 是免疫人工鱼个体的位置向量，对应于均衡器权向量个体。

#### 3. 最优权向量的选择

算法在选取最优个体时具有实时性、随机性以及盲均衡算法要满足迫零条件。因此，最优权向量的选择是将公告板中的最优向量作为自适应频域正交小波多模盲均衡算法的初始权向量。

### 6.4.4 算法仿真

#### 1. βMMA 对不同阶 QAM 信号的均衡实验

**【实验 6.4】** 最小相位水声信道 $\boldsymbol{h}=[0.9656 \quad -0.0906 \quad 0.0578 \quad 0.2368]$；发射信号为 4-QAM、16-QAM、64-QAM、256-QAM，均衡器权长

为 16,信噪比为 25dB,中心抽头初始化。10 次蒙特卡罗仿真结果如图 6.17 所示。

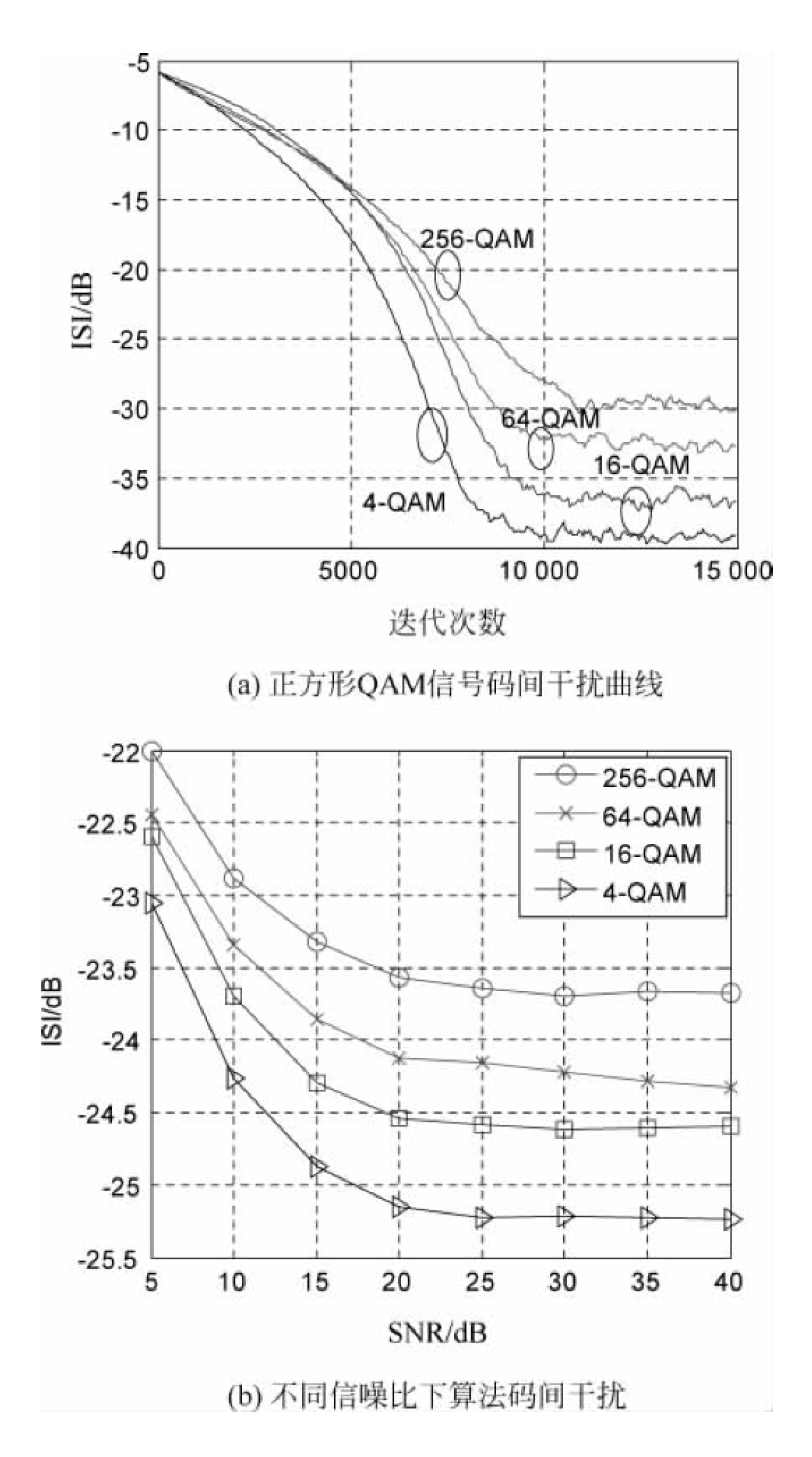



图 6.17　仿真结果

图 6.17 表明,随着 QAM 信号阶数的增加,码间干扰越大且收敛速度越慢;随着信噪比的增大,正方形 QAM 信号的码间干扰趋于减小,且在同一信噪比情况下,调制阶数越低的 QAM 信号码间干扰越小,这说明自适应多模盲均衡算法(βMMA)对低阶 QAM 信号的均衡效果好于高阶;当信噪

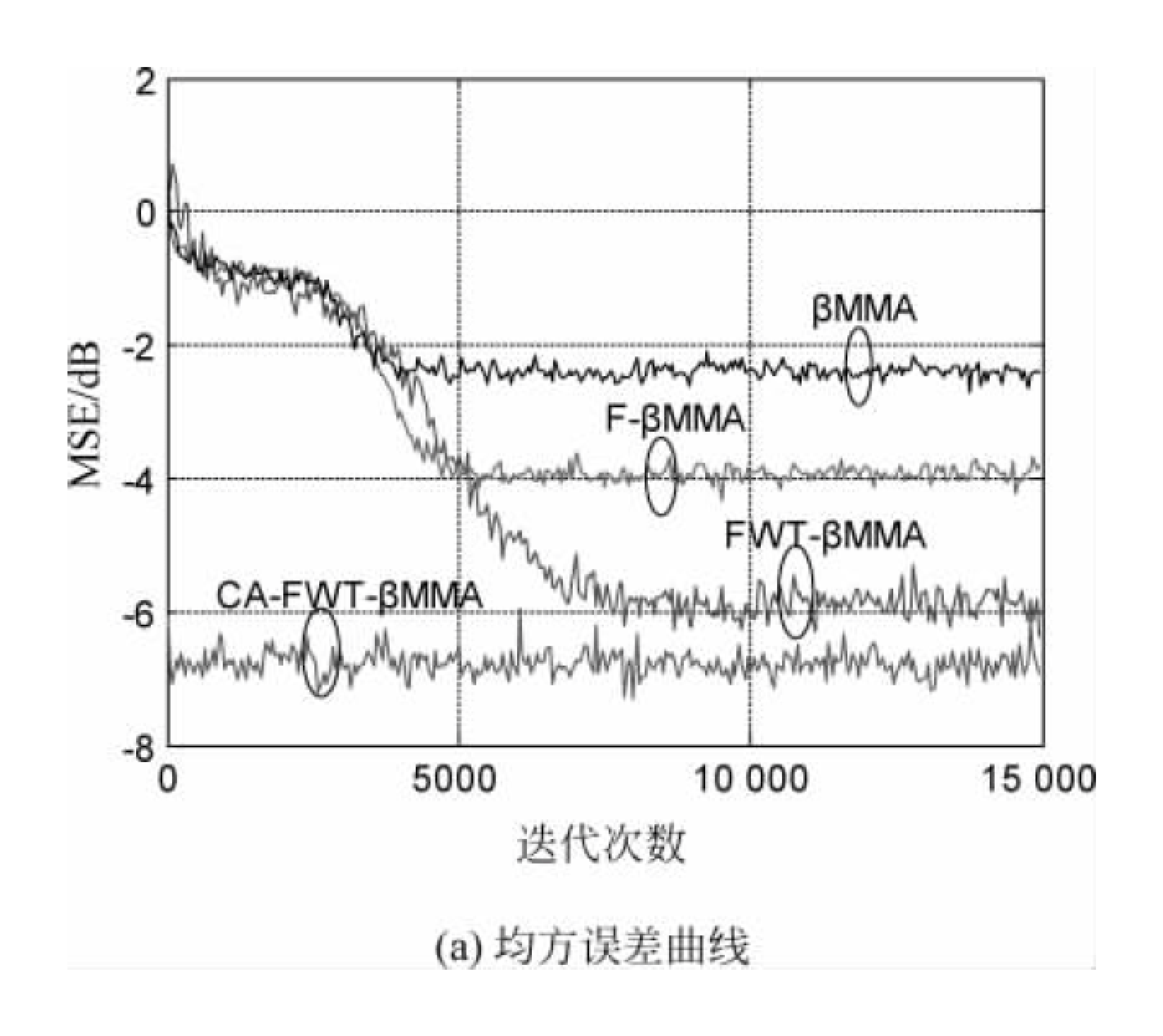
比大于 25 时，码间干扰下降的速率减慢，这说明信噪比在最小相位水声信道中达到一定值时，对均衡器的均衡影响不是很明显。

2. 对 256-QAM 信号的优化实验

**【实验 6.5】** 混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$；发射信号为 256-QAM，均衡器权长为 16，信噪比为 32dB，种群规模为 100，克隆复制控制因子 $p_m=2$，最优交叉概率为 0.2，变异概率为 0.1，人工鱼视野 0.3，步长为 0.1，拥挤度因子为 0.618，最大进化代数为 100，βMMA、F-βMMA 和 FWT-βMMA 均采用中心抽头初始化。其他参数设置，如表 6.4 所示。400 次蒙特卡罗仿真结果如图 6.18 所示。

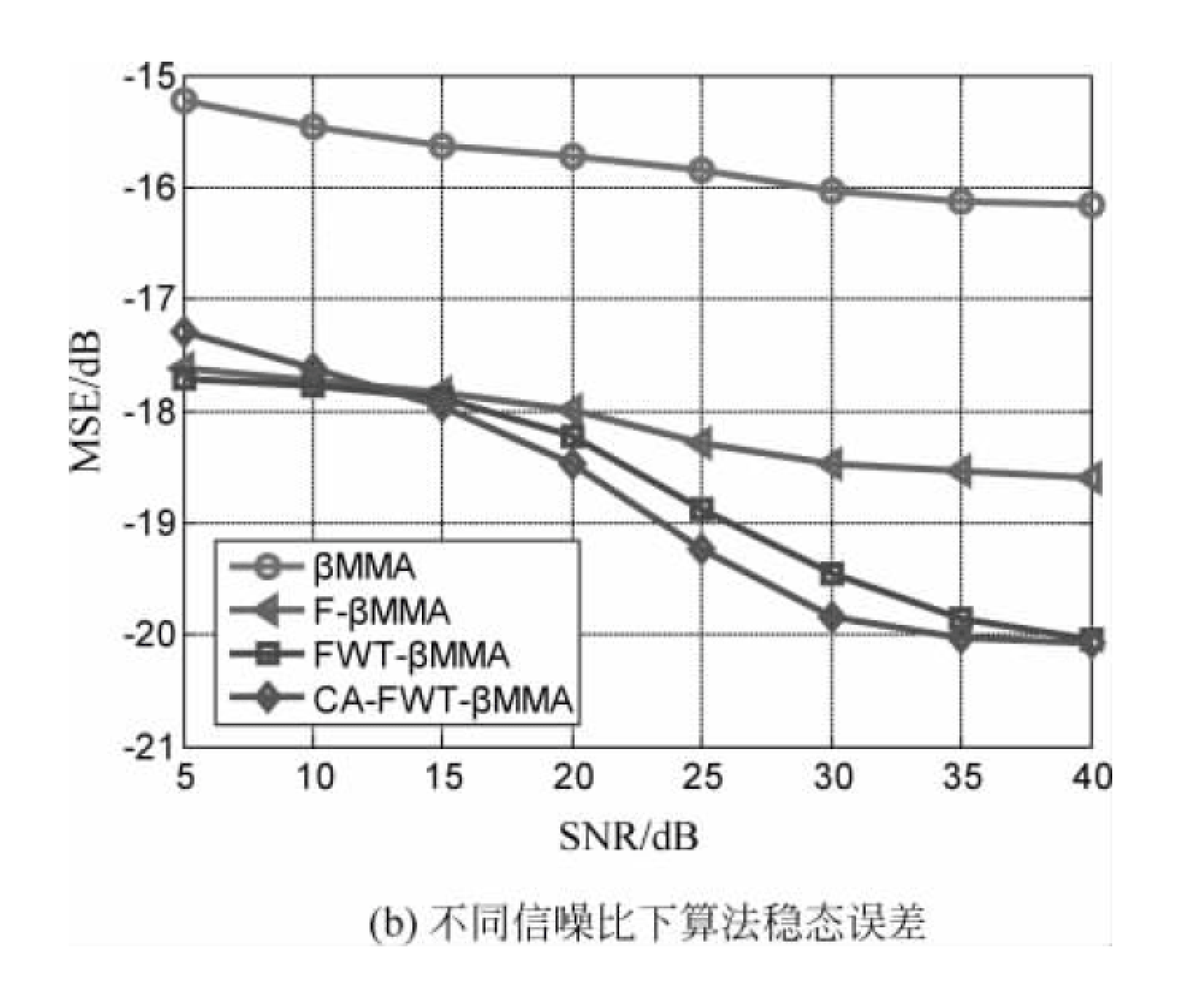
表 6.4 仿真参数设置

| 算 法 | 仿真步长 | 小波 | 分解层数 | 初始化功率 | $\alpha$ 取值 |
|---|---|---|---|---|---|
| βMMA | 0.000 002 | | | | |
| F-βMMA | 0.000 001 6 | | | | |
| FWT-βMMA | 0.000 002 5 | DB2 | 2 | 4 | 0.999 |
| ICAFSA-FWT-βMMA | 0.000 001 | DB2 | 2 | 8 | 0.99 |




(a) 均方误差曲线

图 6.18 仿真结果



(b) 不同信噪比下算法稳态误差

图 6.18 （续）

图 6.18(a)表明，由于高斯噪声的随机性，导致均方误差曲线的抖动较大，但其收敛趋势稳定，这说明算法具有一定的稳定性和适用性；ICAFSA-FWT-βMMA 的稳态误差最小，约为 7dB，比 FWT-βMMA、F-βMMA 和 βMMA 分别减小了 1dB、3dB 和 4.5dB，收敛速度比 FWT-βMMA、F-βMMA 和 βMMA 分别快了近 8000、5000 和 4000 步。图 6.18(b)表明，信噪比越大，均方误差越小，且在同一信噪比的情况下，ICAFSA-FWT-βMMA 的均方误差最小。

## 6.5 基于自适应最小熵的盲均衡算法

为均衡高阶非常模信号和克服相位旋转问题，文献[20]融合常模盲均衡算法计算量小和多模盲均衡算法能纠正相位旋转的优点，给出了一种自适应最小熵盲均衡算法(βCMA)，但收敛速度慢，稳态误差大。

为克服 βCMA 的缺点，本节采用两种方法来加快 βCMA 的收敛速度并降低收敛后的误差。

(1) 将超指数迭代和分别与基于 Delta 势阱、一维谐振子和方势阱的量

子人工鱼群算法引入到自适应最小熵盲均衡算法中。在增加较小计算开销的前提下，利用超指数迭代加快自适应最小熵盲均衡算法的收敛速度，再通过量子人工鱼群算法的快速全局寻优能力，进一步提高收敛速度、减小稳态误差。

（2）采用振幅移相键控（APSK）信号，该信号的星座图有多环结构，能很好地协调信号在传输后的误码性能和发射功率，且能减小码间干扰、加快收敛速度。

### 6.5.1 自适应最小熵盲均衡算法

利用文献[21]原始序列 $\boldsymbol{a}(k)$ 的部分信息，定义了一种新的代价函数，即

$$J(k)=\max_f E[|z(k)|^2] \tag{6.5.1a}$$

$$f_{\max}[R_a,|z(k)|]=R_a \tag{6.5.1b}$$

式(6.5.1b)是式(6.5.1a)的约束条件，$z(k)$为均衡器输出信号；$\boldsymbol{f}$ 为均衡器权向量；$R_a$ 为原始序列的最大幅值。对于 $a,b\in C$，有

$$f_{\max}(|a|,|b|)=\frac{||a|+|b||}{2}+\frac{||a|-|b||}{2}=\begin{cases}|a| & |a|\geqslant|b|\\ |b| & |a|<|b|\end{cases} \tag{6.5.2}$$

若$|z(k)|<R_a$，约束条件(6.5.1b)成立；若$|z(k)|>R_a$，则约束条件(6.5.1b)被违背。为使约束条件满足，利用拉格朗日乘数法将代价函数和约束条件合并为

$$J=\max_f E[|z(k)|^2]+\lambda(f_{\max}[R_a,|z(k)|]-R_a) \tag{6.5.3}$$

$|z(k)|^2$ 和 $f_{\max}(R_a,|z(k)|)$分别对均衡器权向量的求导，得

$$\frac{\partial|z(k)|^2}{\partial \boldsymbol{f}}=\frac{\partial|z(k)|^2}{\partial z(k)}\frac{\partial z(k)}{\partial \boldsymbol{f}} \tag{6.5.4a}$$

$$\frac{\partial f_{\max}(R_a,|z(k)|)}{\partial \boldsymbol{f}}=\frac{\partial f_{\max}(R_a,|z(k)|)}{\partial z(k)}\frac{\partial z(k)}{\partial \boldsymbol{f}}=\frac{g(k)z^*(k)}{4|z(k)|}\boldsymbol{y}(k) \tag{6.5.4b}$$

式中，$g(k)=1+\mathrm{sgn}(|z(k)|-R_a)$；$\boldsymbol{y}(k)$为均衡器输入信号。将式(6.5.3)和式(6.5.4)联立，则权向量更新公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu(1+g(k)/4|z(k)|)z^*(k)\boldsymbol{y}(k) \tag{6.5.5}$$

如果$|z(k)|<R_a$，则$g(k)=0$，权向量更新式(6.5.5)变为$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu z(k)\boldsymbol{y}(k)$；如果$|z(k)|=R_a$，则$g(k)=1$，式(6.5.1a)成立，自适应更新过程停止。为保证$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)$，则$\lambda=-4|z(k)|$；若$|z(k)|>R_a$，则权向量更新式(6.5.5)变为$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)+\mu(1+\lambda/2z(k))\boldsymbol{y}(k)$，约束条件式(6.5.1b)被违背。为使式(6.5.1b)成立，计算$\lambda$，使其满足 Bussgang 过程。令$\lambda=-2(1+\beta)|z(k)|$，则 Bussgang 过程为

$$\underbrace{E[z(k)z^*(k-n)]}_{|z(k)<R_a|}+\underbrace{-\beta E[z(k)z^*(k-n)]}_{|z(k)>R_a|}=0,\quad \forall i\in Z \tag{6.5.6}$$

对于原始序列$\boldsymbol{a}(k)$，长度为$N$且每一个原始信号都落在调制集合$\{R_1,\cdots,R_M\}$中，其中，$R_M=R_a$。$N_l$表示第$l$阶调制信号的数目，$\sum_{l=1}^{M}N_l=N$。对于$n=0$，解式(6.5.6)，得

$$N_1R_1^2+\cdots+N_{M-1}R_{M-1}^2+\frac{1}{2}N_MR_M^2-\frac{\beta}{2}N_MR_M^2=0 \tag{6.5.7}$$

由式(6.5.7)，得

$$\beta=2\frac{N}{N_M}\frac{P_a}{R_a^2}-1 \tag{6.5.8}$$

式中，$P_a=E[|\boldsymbol{a}(k)|^2]$为原始序列的平均能量。将式(6.5.8)代入式(6.5.5)，得自适应最小熵盲均衡算法的权向量更新公式[22]为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu g[z(k)]z^*(k)\boldsymbol{y}(k) \tag{6.5.9a}$$

$$g[z(k)]=\begin{cases}1, & |z(k)|<R_a\\ -\beta, & |z(k)|>R_a\\ 0, & |z(k)|=R_a\end{cases} \tag{6.5.9b}$$

### 6.5.2 超指数自适应最小熵盲均衡算法

超指数迭代算法在均衡器权向量迭代过程中，不断地计算输入信号相

关矩阵的逆矩阵 $\boldsymbol{Q}$，优化步长因子并白化均衡器输入信号，从而加快收敛速度。$\boldsymbol{Q}$ 矩阵的迭代公式为

$$\boldsymbol{Q}(k+1)=\frac{1}{1-\mu_m}\left[\boldsymbol{Q}(k)-\frac{\mu_m\boldsymbol{Q}(k)\boldsymbol{y}^*(k)\boldsymbol{y}^{\mathrm{T}}\boldsymbol{Q}(k)}{1-\mu_m+\mu_m\boldsymbol{y}^{\mathrm{T}}(k)\boldsymbol{Q}(k)\boldsymbol{y}^*(k)}\right] \tag{6.5.10}$$

式中，$\mu_m$ 为矩阵 $\boldsymbol{Q}$ 的迭代步长；$\boldsymbol{Q}$ 的初值为输入信号相关矩阵的逆，即 $\boldsymbol{Q}(0)=\boldsymbol{R}_1^{-1}$。

$$\boldsymbol{R}_1=\boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{C}_1+\frac{\sigma_n^2}{\sigma_a^2}\boldsymbol{I}_L \tag{6.5.11}$$

式中，$\boldsymbol{C}_1$ 为信道的卷积矩阵，$\sigma_a^2$ 和 $\sigma_n^2$ 分别表示噪声和原始序列的方差，$\boldsymbol{I}_L$ 是 $L\times L$ 的单位矩阵。将式(6.5.10)和式(6.5.11)代入式(6.5.9)获得超指数自适应最小熵盲均衡算法(SEI-βCMA)权向量的更新公式为

$$\boldsymbol{f}(k+1)=\boldsymbol{f}(k)-\mu\boldsymbol{Q}(k)g[z(k)]z^*(k)\boldsymbol{y}(k) \tag{6.5.12}$$

### 6.5.3 基于量子人工鱼群算法优化的超指数自适应最小熵盲均衡算法

#### 1. 基本 AFSA 模型

在 $D$ 维搜索空间中，$N$ 条人工鱼组成一个种群。其中 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{iD})$ 表示第 $i$ 条人工鱼的位置向量，人工鱼当前位置的食物浓度为 $Y=\mathrm{Fit}(X)$；两条鱼间的距离为 $d_{i,j}=\|\boldsymbol{X}_i-\boldsymbol{X}_j\|$；感知距离、最大移动步长、拥挤度因子分别为 Visual，step 和 $\delta$。人工鱼通过觅食、聚群及追尾等行为来更新位置，位置向量更新公式分别为

$$\boldsymbol{X}_j=\boldsymbol{X}_i+(2r-1)\cdot\mathrm{Visual}\cdot\boldsymbol{e}_i \tag{6.5.13}$$

$$\boldsymbol{X}_j=\boldsymbol{X}_i+r\cdot\mathrm{step}\cdot\frac{\boldsymbol{X}_c-\boldsymbol{X}_i}{\|\boldsymbol{X}_c-\boldsymbol{X}_i\|} \tag{6.5.14}$$

$$\boldsymbol{X}_j=\boldsymbol{X}_i+r\cdot\mathrm{step}\cdot\frac{\boldsymbol{X}_{\mathrm{gbest}}-\boldsymbol{X}_i}{\|\boldsymbol{X}_{\mathrm{gbest}}-\boldsymbol{X}_i\|} \tag{6.5.15}$$

式中，$r$ 为(0,1)之间的随机数，为创造力系数；$\boldsymbol{e}_i$ 为单位向量。

2. 量子势阱 AFSA 模型

在量子理论中，量子的动态行为用薛定谔方程描述。

$$\mathrm{j}h\frac{\partial}{\partial t}\psi(r,t)=\left(-\frac{h^2}{2m'}\nabla^2+V(r)\right)\psi(r,t) \tag{6.5.16}$$

式中，$m'$为量子的质量，$V(r)$为势能分布函数，$h$ 为普朗克常量。在薛定谔方程中，根据波函数 $\psi(r,t)$的统计诠释，它的幅值的平方为量子在某一位置出现的概率。

人工鱼群在量子势阱的行为表示，称为量子人工鱼群（quantum artificial fish school algorithm，QAFSA）。QAFSA 中人工鱼在势场中出现的概率密度函数为$|\psi(r)|^2$，通过设计合理的势阱参数来控制量子人工鱼的行为。首先选择某种不含时间 $t$ 的势阱，然后通过求解薛定谔方程分离出波函数 $\psi(r)$，获人工鱼群迭代中的随机数 $r$。现参照文献[23]量子势阱粒子算法原理，构建 Delta 势阱、一维谐振子和方势阱人工鱼，并说明量子人工鱼群算法（QAFSA）的构造过程。

（1）Delta 势阱 AFSA 模型

Delta 势阱的势能分布为

$$V(r)=-\gamma\delta(r) \tag{6.5.17}$$

式中，$\gamma$ 为势阱深度。联立式(6.5.16)和式(6.5.17)，得人工鱼波函数为

$$\psi(r)=\frac{1}{\sqrt{L}}\mathrm{e}^{-|r|/L} \tag{6.5.18}$$

式中，$L=\dfrac{h^2}{m'\gamma}$为 Delta 势阱的特征长度。人工鱼在 $r$ 处出现的概率密度函数为

$$Q(r)=|\psi(r)|^2=\frac{1}{L}\mathrm{e}^{-2|r|/L} \tag{6.5.19}$$

为使人工鱼在 $r$ 处以较大的概率向最优位置移动，式(6.5.19)应满足的关系为

$$\int_{-|r|}^{|r|}Q(r)\mathrm{d}r>0.5 \tag{6.5.20}$$

由式(6.5.19)和式(6.5.20),得特征长度为

$$L = \frac{|r|}{g\ln(\sqrt{2})} \tag{6.5.21}$$

式中,$g$ 为控制参数且 $g>1$。

在任一确定时刻,势阱中人工鱼的位置是不确定的,而普通人工鱼的位置是确定的,为克服该矛盾,可借助波函数塌陷解决。

令 $u=\mathrm{e}^{-2|r|/L}$,且 $u\subset(0,1)$,得

$$|r| = \frac{L}{2}\ln(1/u) \tag{6.5.22}$$

联立式(6.5.21)和式(6.5.22),得

$$r(k+1) = \frac{\ln(1/u)}{2g\ln\sqrt{2}}r(k) \tag{6.5.23}$$

由以上分析,称 $r(k)$ 为创造力系数,这时量子 Delta 势阱人工鱼群算法(quantum delta potential well artificial fish swarm algorithm,QDAFSA)中人工鱼位置向量的更新公式为

$$\boldsymbol{X}_j = \boldsymbol{X}_i + \left(2\frac{\ln(1/u)}{2g\sqrt{2}} - 1\right)\cdot \mathrm{Visual}\cdot \mathbf{one}(1,N) \tag{6.5.24}$$

$$\boldsymbol{X}_j = \boldsymbol{X}_i + \frac{\ln(1/u)}{2g\sqrt{2}}\cdot \mathrm{step}\cdot \frac{\boldsymbol{X}_c - \boldsymbol{X}_i}{\|\boldsymbol{X}_c - \boldsymbol{X}_i\|} \tag{6.5.25}$$

$$\boldsymbol{X}_j = \boldsymbol{X}_i + \frac{\ln(1/u)}{2g\sqrt{2}} * \mathrm{step} * \frac{\boldsymbol{X}_{gbest} - \boldsymbol{X}_i}{\|\boldsymbol{X}_{gbest} - \boldsymbol{X}_i\|} \tag{6.5.26}$$

(2) 一维谐振子和方势阱 AFSA 模型

一维谐振子和方势阱的势能分别为

$$V(r) = \frac{1}{2}Kr^2 \tag{6.5.27}$$

$$V(r) = \begin{cases} 0, & |r| \leqslant W/2 \\ V_0, & |r| > W/2 \end{cases} \tag{6.5.28}$$

式中,$K$ 为描述简谐作用力强弱的参数;$W$ 为势陷阱宽度;$V_0$ 为势阱高度。由文献[23],得一维谐振子和方势阱的 $r(k+1)$ 分别为

$$|r(k+1)|_{QO} = \frac{\sqrt{\ln(1/u)}}{0.4769g}|r(k)| \tag{6.5.29}$$

$$|r(k+1)|_{QS} = \frac{1.4829\arccos(\sqrt{u})}{g}|r(k)| \tag{6.5.30}$$

将式(6.5.13)～式(6.5.15)中的创造力系数 $r$ 用式(6.5.29)和式(6.5.30)代替，就得一维谐振子(quantum one-dimensional harmonic oscillator，QO)和方势阱(quantum square potential well，QS)人工鱼群位置向量的更新公式为

$$\boldsymbol{X}_j = \boldsymbol{X}_i + (2H-1)\cdot \text{Visual}\cdot \mathbf{one}(1,N) \tag{6.5.31}$$

$$\boldsymbol{X}_j = \boldsymbol{X}_i + H\cdot \text{step}\cdot \frac{\boldsymbol{X}_c - \boldsymbol{X}_i}{\|\boldsymbol{X}_c - \boldsymbol{X}_i\|} \tag{6.5.32}$$

$$\boldsymbol{X}_j = \boldsymbol{X}_i + H\cdot \text{step}\cdot \frac{\boldsymbol{X}_{\text{gbest}} - \boldsymbol{X}_i}{\|\boldsymbol{X}_{\text{gbest}} - \boldsymbol{X}_i\|} \tag{6.5.33}$$

式中，$H$ 表示 $\sqrt{\ln(1/u)}/0.4769g$ 或 $1.4829\arccos(\sqrt{u})/g$。

3. 算法流程

对于一个量子人工鱼群系统，人工鱼当前位置和食物浓度信息不能同时确定，因此将其作为一个参数。基于 QAFSA 优化均衡器权向量的流程[24]如下。

步骤 1：量子人工鱼参数初始化。设人工鱼的种群数目为 $N$，每条人工鱼的维数为 $D$，初始化人工鱼的位置向量 $\boldsymbol{X}_i=(x_{i1},x_{i2},\cdots,x_{iD})$。

步骤 2：计算适应度函数(食物浓度)。根据式(6.5.3)获得盲均衡算法的代价函数。由于盲均衡算法的目的是将代价函数迭代至最小值，而 QAFSA 的目的是寻找食物浓度最大的位置。为克服该矛盾，将代价函数的倒数定义为 QAFSA 的适应度函数，即

$$\text{Fit}(\boldsymbol{X}_i) = \frac{1}{J_{\beta\text{CMA}}(\boldsymbol{X}_i)} \tag{6.5.34}$$

步骤 3：初始化公告板。计算每条人工鱼当前状态 $\boldsymbol{X}_i$ 下的适应度 $\text{Fit}(\boldsymbol{X}_i)$，将最优的人工鱼位置向量和适应度值作为公告板的初始值

$\boldsymbol{X}_{\text{best}}$和 $\text{Fit}_{\text{best}}$。

步骤 4：算法更新。对每条量子人工鱼进行觅食、聚群、追尾等行为；将得到的最优值与公告板比较，若优于公告板，则更新公告板；否则，跳转步骤 5。

步骤 5：终止条件判断。判断算法是否达到最大进化次数，若小于进化次数，跳转至步骤 4；若满足结束条件，则终止。

步骤 6：均衡器权向量初始化。将由量子人工鱼群算法得到的最优位置向量，作为均衡器的初始权向量。

### 6.5.4 基于 APSK 调制信号的自适应最小熵盲均衡算法

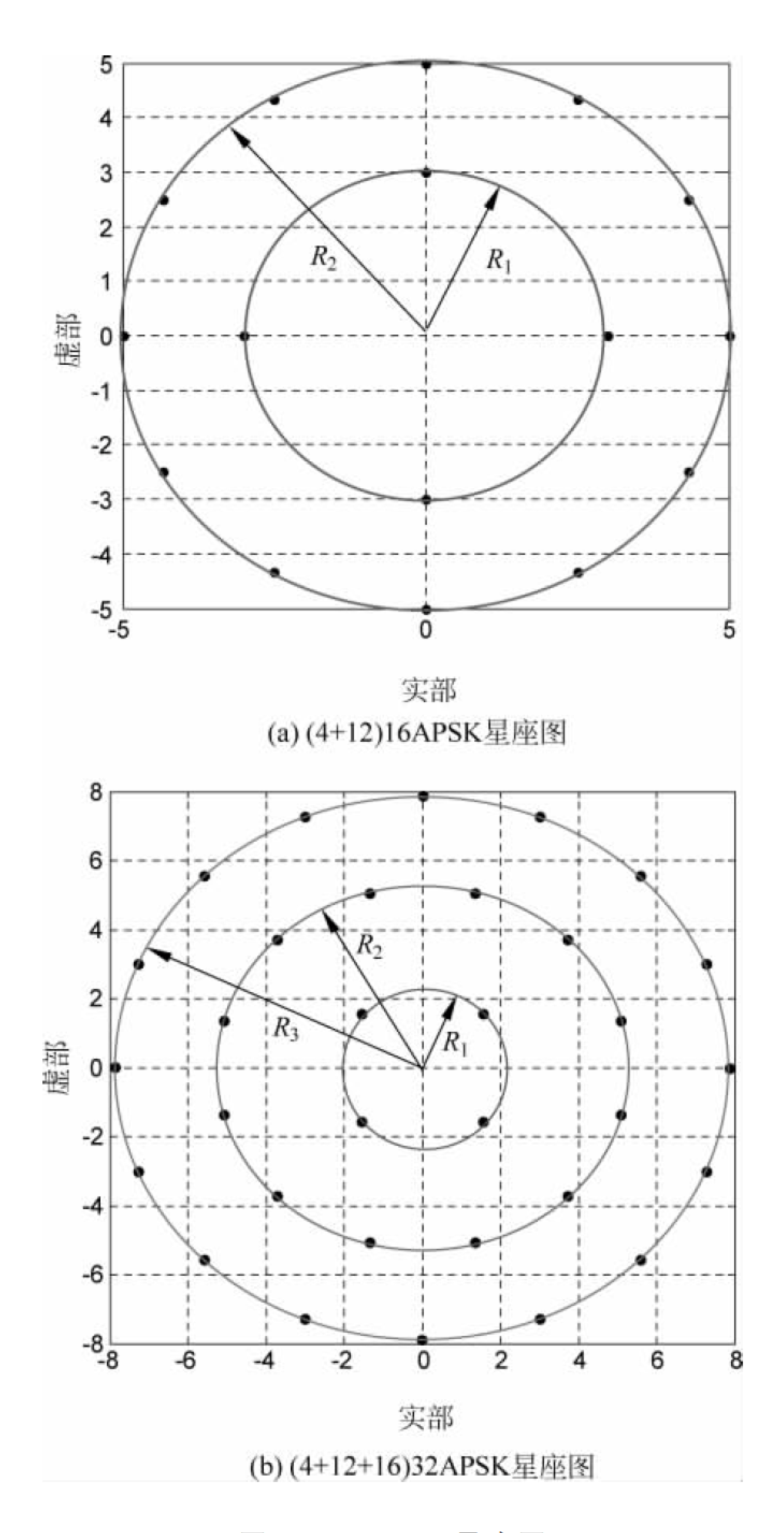
#### 1. APSK 调制信号[25-26]

欧洲 ESTI 制定了新一代数字视频广播标准——DVS-S2 标准，该标准中多进制振幅移相键控(M-APSK)信号越来越引人瞩目。该调制方式既能提高频带利用率又能增加功率有效性，适合于信号在复杂信道中传输。M-APSK 是一种利用载波相位和幅度同时承载信息的混合调制方式，其星座图成圆形且圆周个数少。

M-APSK 是由 $K$ 个同心圆构成，每个同心圆上有等间距的 PSK 信号点。M-APSK 星座图的信号集为 $X$ 且每个信号点都是复数，其表达式为

$$X=\begin{cases} r_1\exp\left(\mathrm{j}\left(\dfrac{2\pi}{n_1}i+\theta_1\right)\right), & i=0,1,\cdots,n_1-1 \\ r_2\exp\left(\mathrm{j}\left(\dfrac{2\pi}{n_2}i+\theta_2\right)\right), & i=0,1,\cdots,n_2-1 \\ \vdots & \\ r_K\exp\left(\mathrm{j}\left(\dfrac{2\pi}{n_K}i+\theta_K\right)\right), & i=0,1,\cdots,n_K-1 \end{cases} \tag{6.5.35}$$

式中，$n_l$、$r_l$ 和 $\theta_l(l=1,2,\cdots,K)$分别表示第 $l$ 个圆上的信号点数、半径和信号点的相对相移；$\mathrm{j}=\sqrt{-1}$。为充分利用星座图的欧式空间，应满足条件 $n_l<n_{l+1}$，即外圆的信号点数要大于内圆。基于这样的分析，M-APSK 信号可写为$(n_1+\cdots+n_K)$APSK 的形式，如图 6.19 所示。



(a) (4+12)16APSK星座图

(b) (4+12+16)32APSK星座图

图 6.19　APSK 星座图

在 DVB-S2 标准中，采用了 $X=(4+12)$ APSK 和 $X=(4+12+16)$ APSK，分别有 $n_K=2$ 和 $n_K=3$ 个圆。同时，对信号能量作归一化处理，即 $\sum_{l=1}^{K} n_l r_l^2 = M$(其中，$M$ 为星座图中信号的总数)。

M-APSK 星座图设计的参数：每个圆的半径 $r_l$、圆环上星座点的数目 $n_l$、星座点的相对相位偏移 $\theta_l$。

2. 基于 APSK 调制信号的 βCMA 系统

编码调制盲均衡系统模型，如图 6.20 所示。

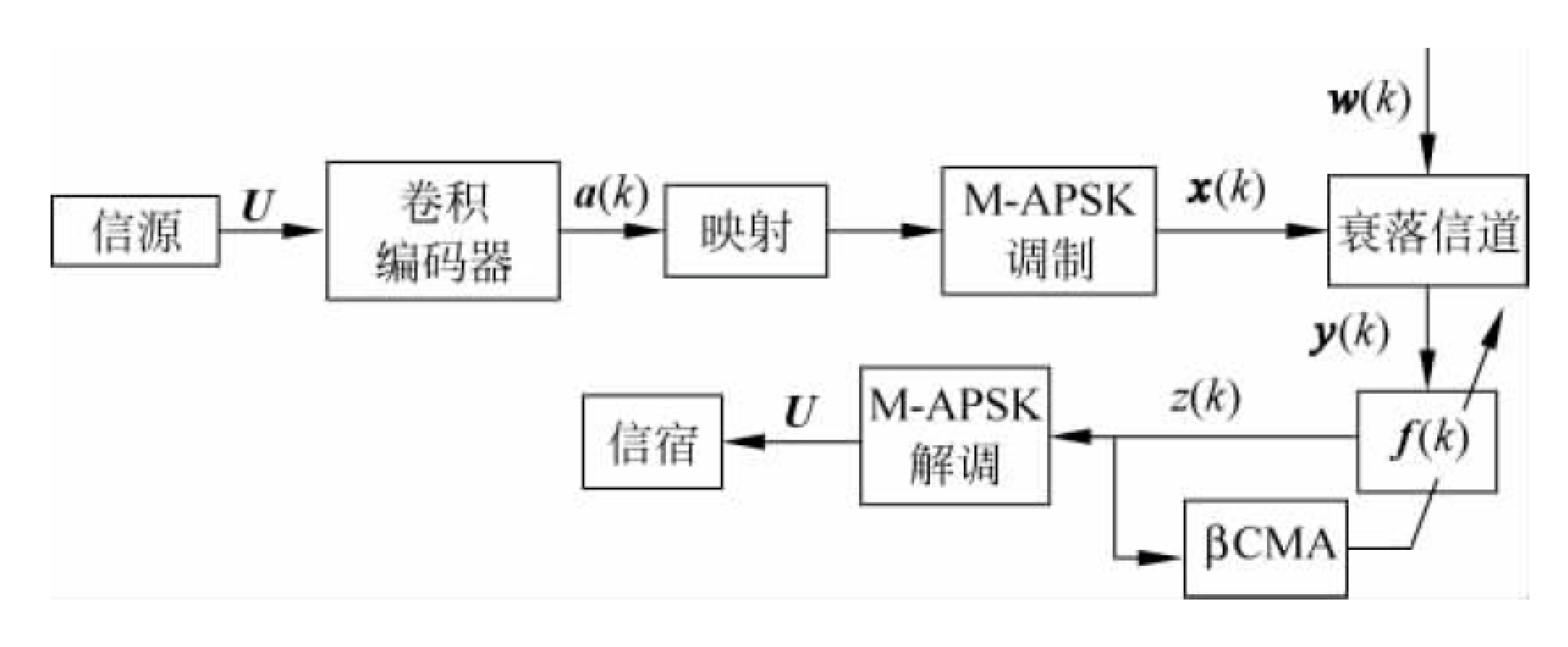



图 6.20 编码调制盲均衡系统模型

假设 $\boldsymbol{U}$ 是从发射机发射的二进制序列，$\boldsymbol{a}(k)$是经过卷积编码器的输出信号。图中的映射部分把二进制序列转化为调制器所需的多电平序列。将映射后的信号通过 M-APSK 调制器，进行 APSK 调制。调制后的信号经过衰落信道后，利用盲均衡器对失真信号进行均衡以恢复 APSK 已调信号，最后通过解码器还原成原始的信源信号。其中，$\boldsymbol{w}(k)$是均值为 0、方差为 1 的加性高斯白噪声；$\boldsymbol{x}(k)$和 $\boldsymbol{y}(k)$分别为信道和均衡器的输入序列；$z(k)$为均衡器输出序列。

### 6.5.5 算法仿真

以 βCMA、SEI-βCMA 为比较对象，对基于量子 Delta 势阱、一维谐振子和方势阱人工鱼群算法优化的超数自适应最小熵盲均衡算法（依次记为 QD-SEI-βCMA，QO-SEI-βCMA 和 QS-SEI-βCMA）性能进行验证。

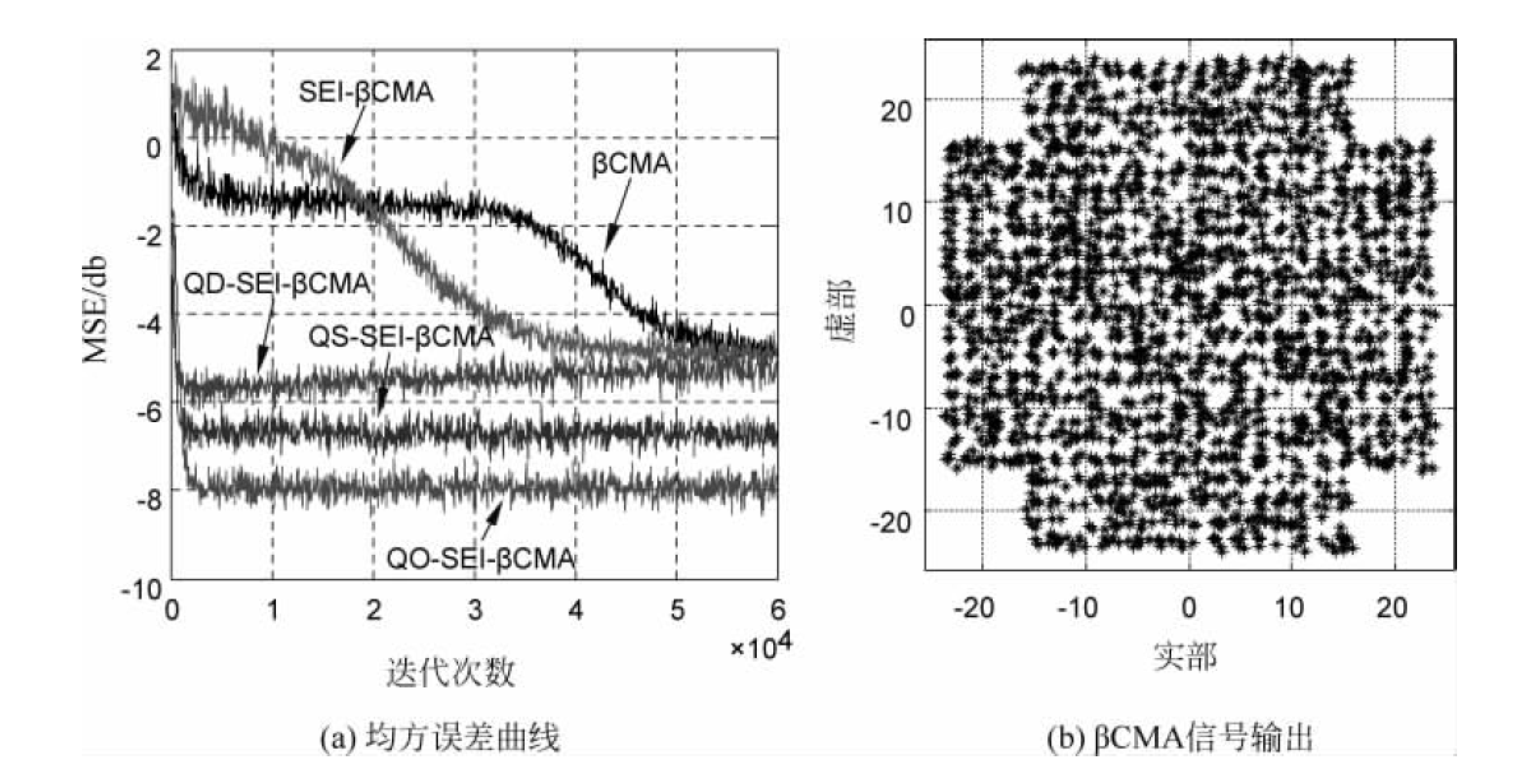
1. 512-QAM 信号的优化实验

**【实验 6.6】** 最小相位水声信道 $\boldsymbol{h}=[0.9656 \quad -0.0906 \quad 0.0578 \quad 0.2368]$，发射信号为 512-QAM，信噪比为 44dB，均衡器权长为 16，βCMA

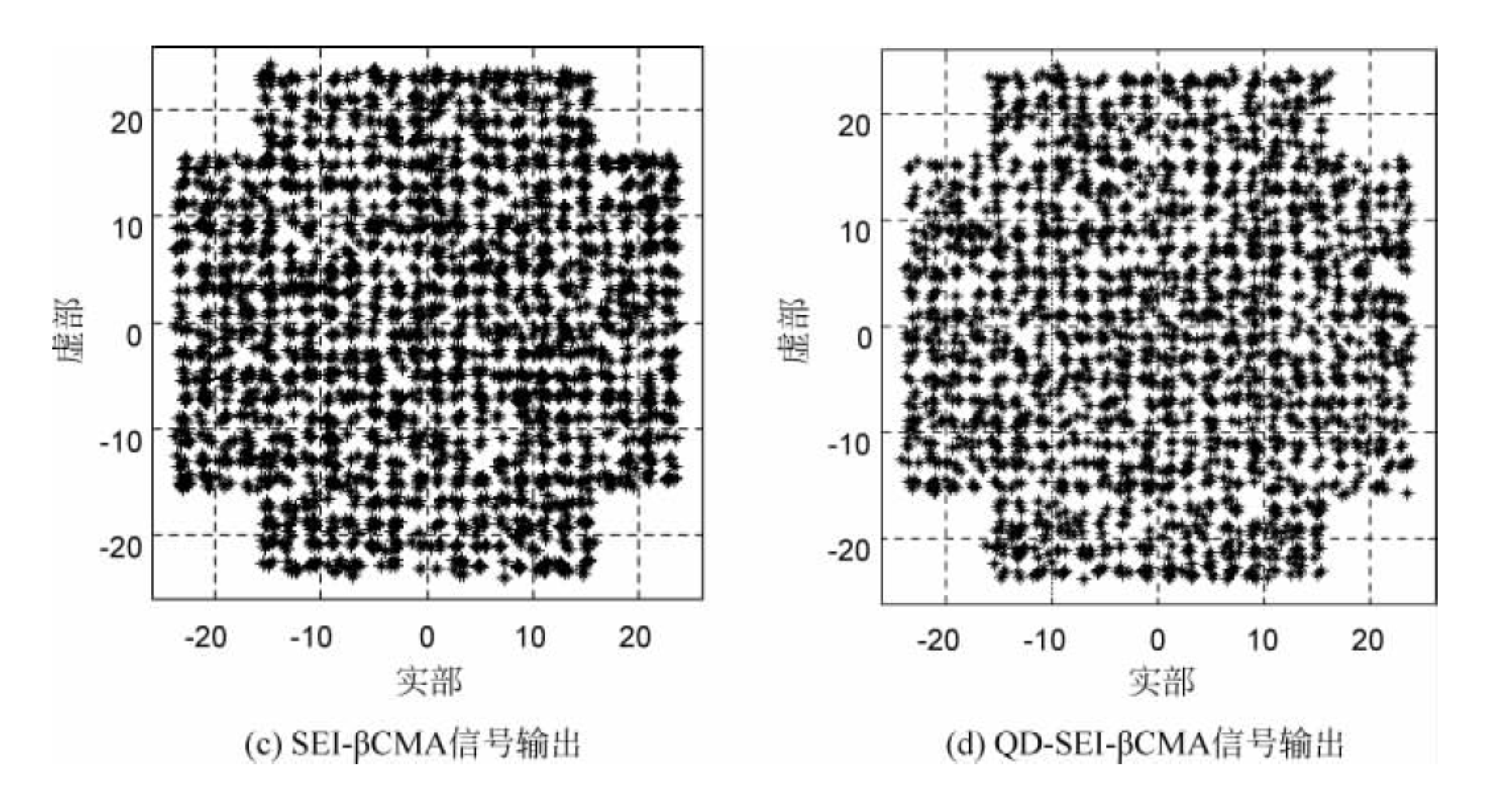
和 SEI-βCMA 采用中心抽头初始化。量子势阱人工鱼群规模为 100，步长 step、视野范围 Visual 和拥挤度因子 $\delta$ 分别取 0.75、0.3 和 0.618。迭代步长参数，如表 6.5 所示。400 次蒙特卡罗仿真结果如图 6.21 所示。

表 6.5　步长参数选择

| | βCMA | SEI-βCMA | QD-SEI-βCMA | QO-SEI-βCMA | QS-SEI-βCMA |
|---|---|---|---|---|---|
| 步长 | $2\times10^{-6}$ | $2.5\times10^{-7}$ | $1\times10^{-8}$ | $2.1\times10^{-8}$ | $8\times10^{-9}$ |
| SEI 步长 | | $1\times10^{-5}$ | $6\times10^{-6}$ | $4\times10^{-6}$ | $5\times10^{-6}$ |



(a) 均方误差曲线　　(b) βCMA信号输出



(c) SEI-βCMA信号输出　　(d) QD-SEI-βCMA信号输出

图 6.21　仿真结果

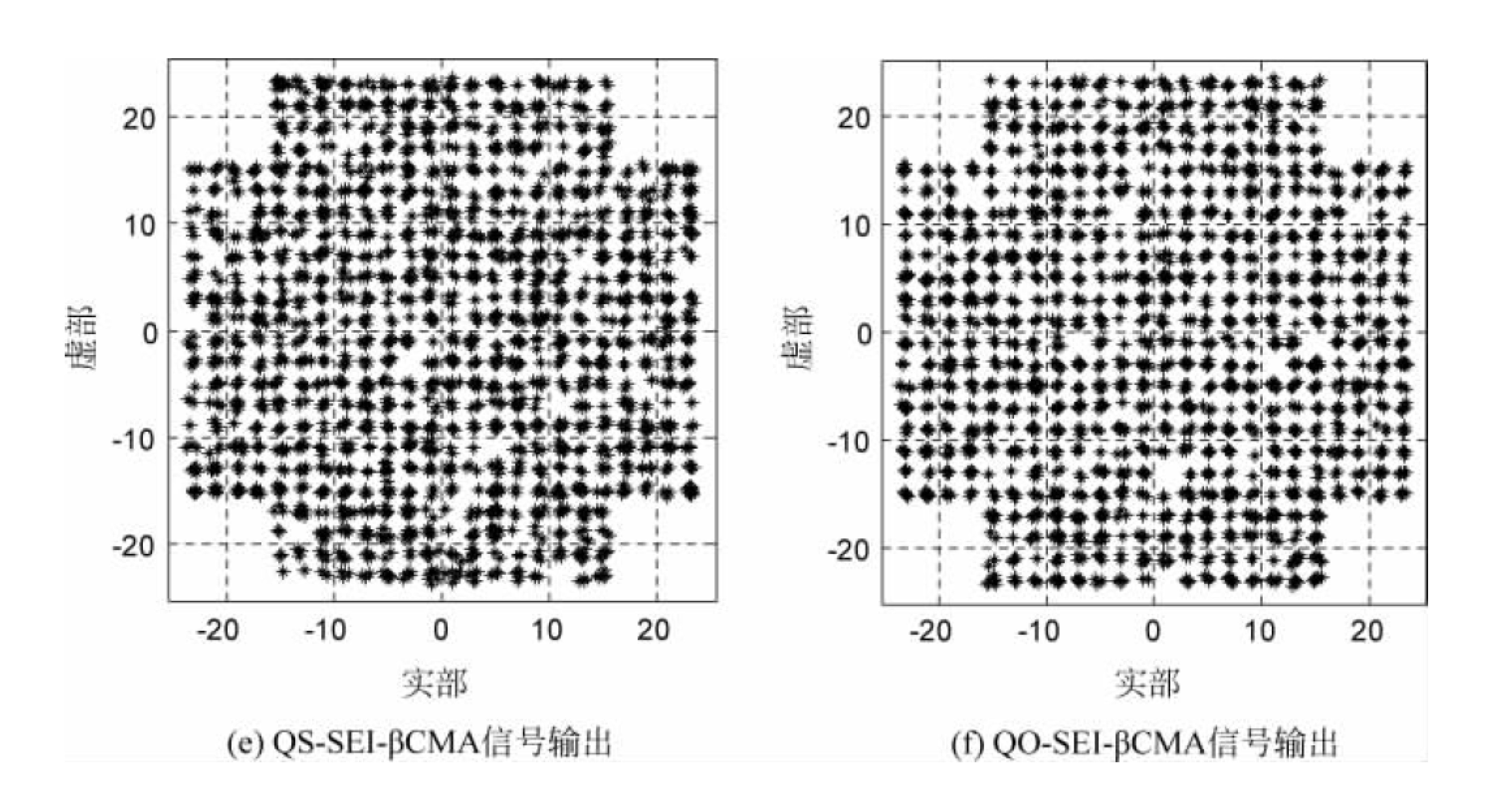



图 6.21 （续）

图 6.21 表明，QO-SEI-βCMA、QO-SEI-βCMA 和 QD-SEI-βCMA 在收敛速度方面几乎相同，比 SEI-βCMA 和 βCMA 分别快约 40 000 步和 50 000 步；在稳态误差方面，QO-SEI-βCMA 较 QS-SEI-βCMA、QD-SEI-βCMA、SEI-βCMA 和 βCMA 分别减小了约 1dB、2.5dB、3dB 和 3.5dB。且 QO-SEI-βCMA 的眼图最为清晰、紧凑。

2. βCMA 对不同调制信号的均衡实验

**【实验 6.7】** 两径相位水声信道 $\boldsymbol{h}=[0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$，发射信号为 16-PSK、16-APSK、16-QAM，信噪比为 20dB，均衡器权长为 16，中心抽头初始化，PSK、APSK、QAM 的步长分别取 $3\times10^{-4}$、$6\times10^{-3}$、$5\times10^{-5}$。30 次蒙特卡罗仿真结果如图 6.22 所示。

图 6.22 表明，对 APSK 的码间干扰较 PSK 和 QAM 分别减少了约 3dB 和 10dB；这说明在相同的信噪比下，对 APSK 的均衡效果最好，且随着信噪比的增加，码间干扰呈减小趋势；在不同信噪比下，APSK 的码间干扰最小。APSK 的眼图较其他调制方式更加清晰、紧凑。

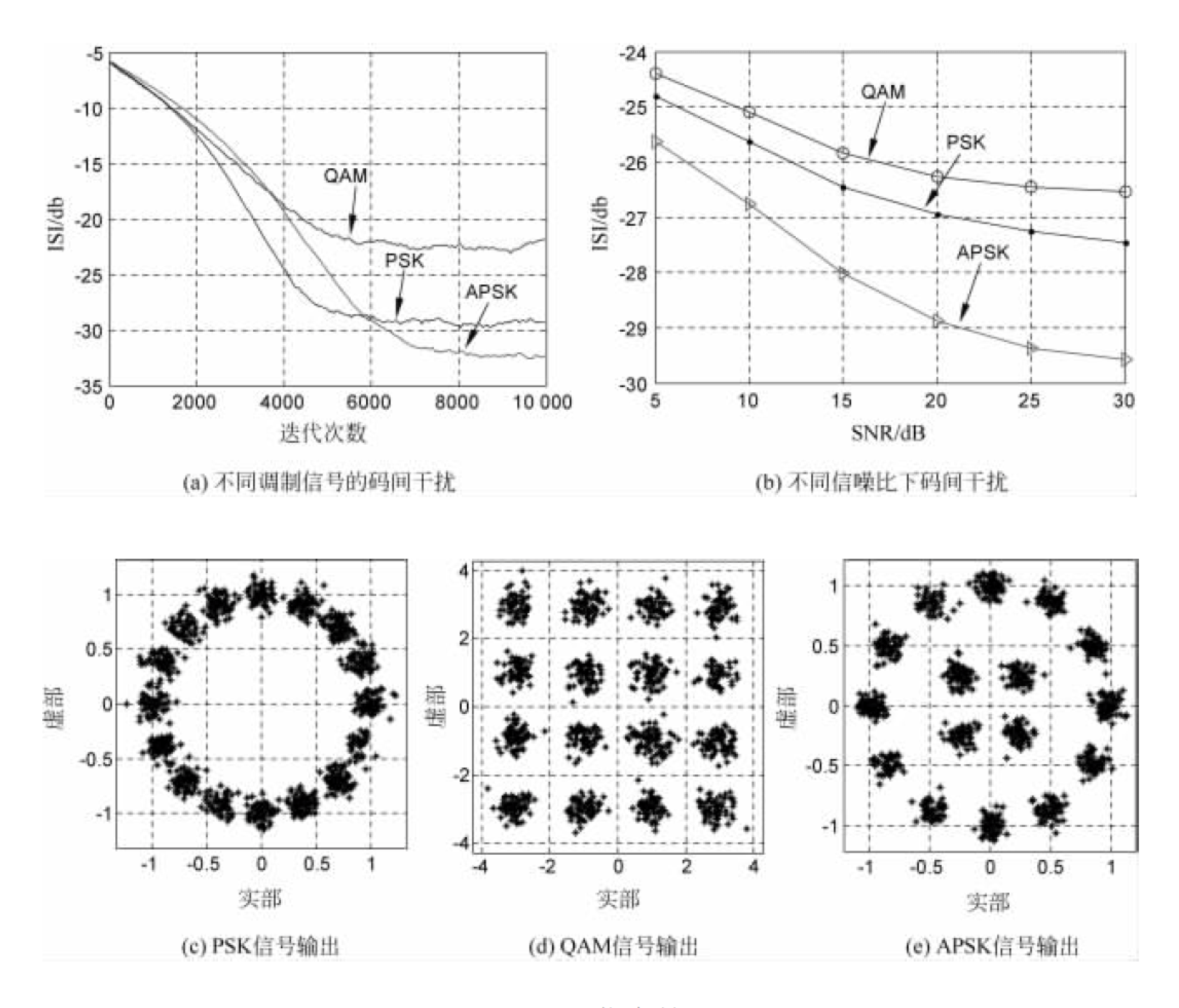



图 6.22 仿真结果

3. 不同进制和圆环数的 APSK 均衡实验

**【实验 6.8】** 混合相位水声信道 $\boldsymbol{h}=[0.3576-0.1611i \quad 0.0227 \quad 0.2645-0.100i, 0.2922-0.4230i]$其零点和频率分布，如图 6.23 所示。信噪比为 25dB，均衡器权长为 16，中心抽头初始化。发射信号分别为(4+12)16-APSK、(6+10)16-APSK、(4+12+16)32-APSK、(6+12+14)32-APSK、(4+6+8+14)32APSK，其步长分别取 $7\times10^{-6}$、$1.4\times10^{-5}$、$4\times10^{-5}$、$2\times10^{-5}$、$6.5\times10^{-7}$。30 次蒙特卡罗仿真结果如图 6.24 所示。

图 6.24(a)～(c)表明：16-APSK 调制信号较 32-APSK 的均衡效果在码间干扰、均方误差方面更优越，且圆环数目多的均衡效果好。图 6.25(e)～(i)表明，βCMA 对 APSK 调制信号均具有良好的均衡性能。

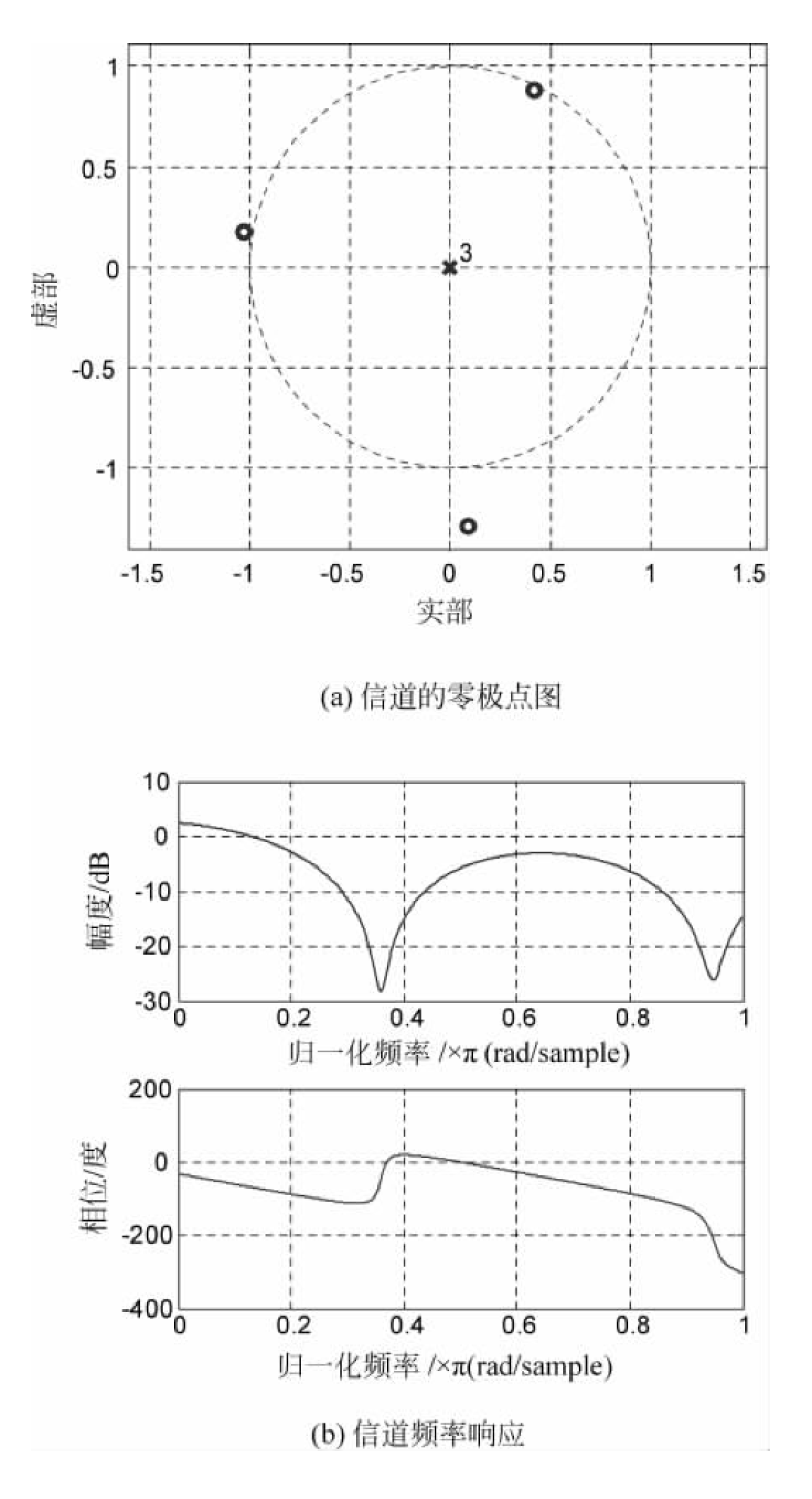
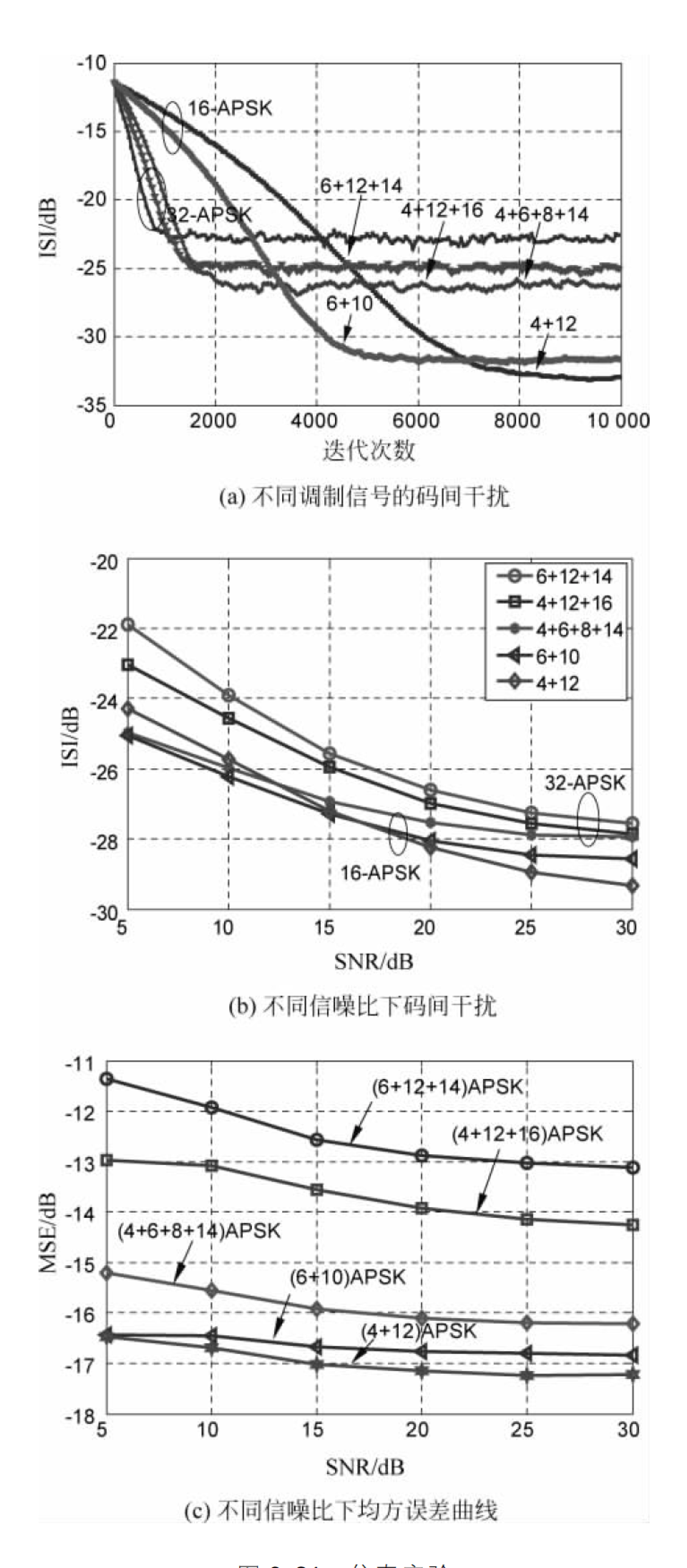

1
0.5
0
-0.5
-1
3
虚部
-1.5
-1
-0.5
0
0.5
1
1.5
实部
(a) 信道的零极点图
10
0
-10
-20
-30
幅度/dB
0
0.2
0.4
0.6
0.8
1
归一化频率 /×π (rad/sample)
200
0
-200
-400
相位/度
0
0.2
0.4
0.6
0.8
1
归一化频率 /×π(rad/sample)
(b) 信道频率响应

图 6.23　信道特性




(a) 不同调制信号的码间干扰

(b) 不同信噪比下码间干扰

(c) 不同信噪比下均方误差曲线

图 6.24 仿真实验

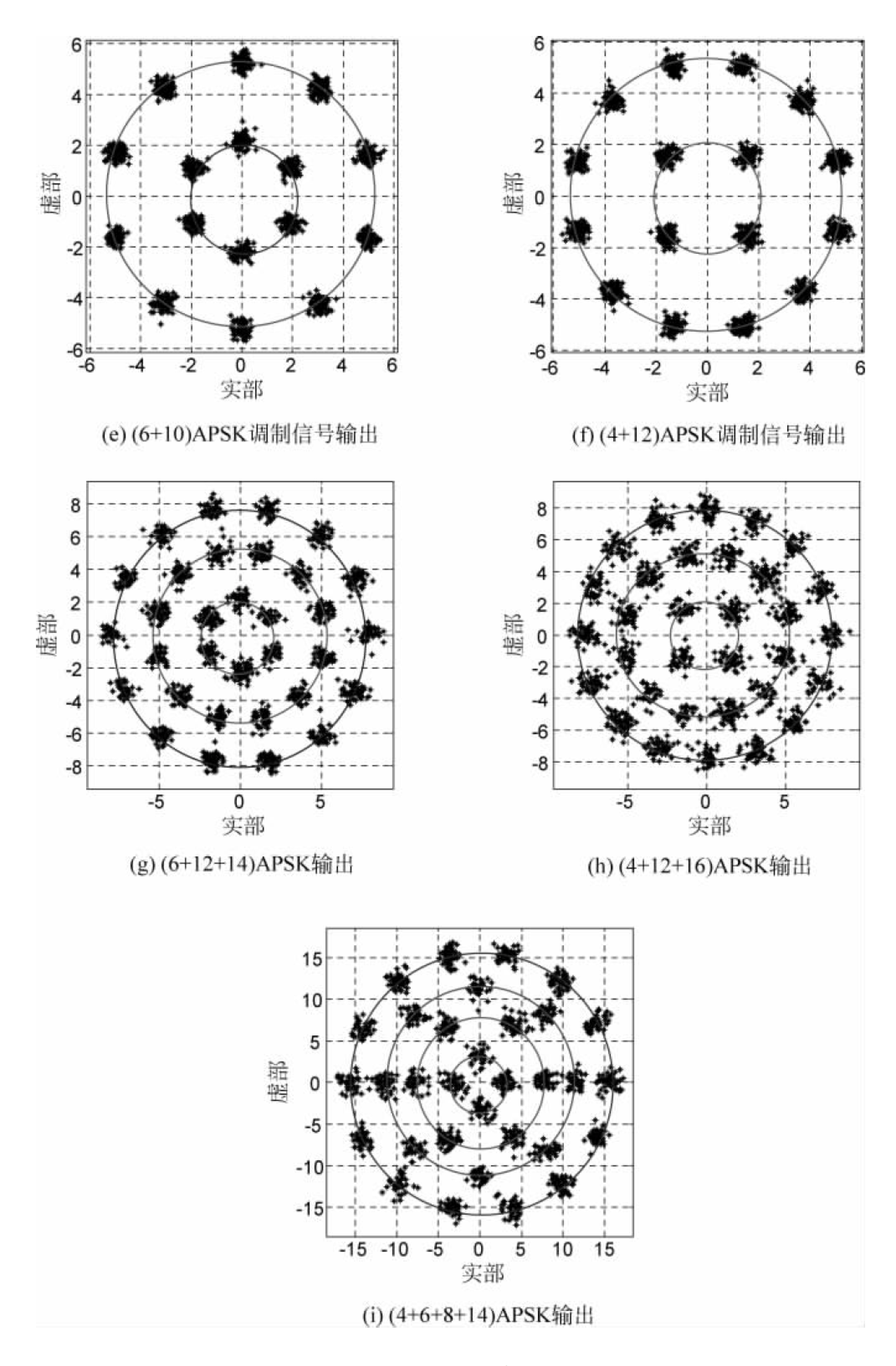

虚部
实部
(e) (6+10)APSK调制信号输出
(f) (4+12)APSK调制信号输出
(g) (6+12+14)APSK输出
(h) (4+12+16)APSK输出
(i) (4+6+8+14)APSK输出

图 6.24 （续）

# 参考文献


［1］ 李晓磊. 一种新型的智能优化算法-人工鱼群算法［D］. 杭州：浙江大学，2003.

［2］ 李晓磊，钱积新. 基于分解协调的人工鱼群优化算法研究［J］. 电路与系统学报，2003，8(1)：1-6.

［3］ 王联国，洪翼. 基于冯·诺依曼邻域结构的人工鱼群算法［J］，控制理论与应用，2010，27(6)：775-780.

［4］ 王联国. 人工鱼群算法及其应用研究［D］. 兰州：兰州理工大学，2009.

［5］ 张梅凤，邵诚. 基于变异算子与模拟退火混合的人工鱼群优化算法［J］. 电子学报，2006，34(8)：1381-1385.

［6］ 张德富，彭煜. 求解三维装箱问题的混合模拟退火算法［J］. 计算机学报，2009，32(11)：2147-2156.

［7］ 黄伟，郭业才. 模拟退火与人工鱼群变异优化的小波盲均衡算法［J］. 计算机应用研究，2012，29(12)：4124-4126.

［8］ Yang C，Guo Y C，Zhu J. Super-exponential iterative blind equalization algorithm based on orthogonal wavelet packer transform［C］. Proceedings of the 9th International Conference on Signal Processing，Beijing. 2008，10：26-29，1830-1833.

［9］ 曲良东，何登旭. 求解不动点方程的人工鱼群算法［J］. 计算机应用研究，2010，27(9)：3356-3357.

［10］ Li X L，Zhang X D. A family of generalized constant modulus algorithm for blind equalization［J］. IEEE Transaction on Communication，2006，54(11)：1913-1917.

［11］ 窦高齐，高俊. 适合于高阶 QAM 系统的多模盲均衡新算法［J］. 电子与信息学报，2008，30(2)：388-391.

［12］ Jian Yang，Jean-Jacques，Guy A. Dumont. The multimodulus blind equalization and Its generalized algorithms［J］. IEEE Journal on Selected Areas in Communications，2002，20(5)：997-1015.

［13］ Zhang Y B，Zhao W J. Blind adaptive MMMSE equalization of underwater acoustic channels based on the linear prediction method［J］. Journal of Marine Science and Application，2011，10(1)：113-120.

［14］ Jian Y，Jean J W. The multimodulus equalization and its generalized algorithm［J］.

IEEE Journal on Selected Areas in Communications,2002,20(5):997-1015.

[15] 黄伟.基于人工鱼群优化的小波盲均衡算法[D].淮南:安徽理工大学,2013.

[16] 何成兵,黄建国,孟庆微,等.基于扩频码的单载波迭代频域均衡水声通信[J].物理学报,2013,62(23):234-301.

[17] Meng C, Tuqan J, Ding Z. A Quadratic programming approach to blind equalization and signal separation[J]. IEEE Transaction on Signal Processing, 2009,57(6):232-2244.

[18] Shafyat A, Asoke K N. An adaptive constant modulus blind equalization and its stochastic stability analysis[J]. IEEE Signal Processing Letters, 2010, 17(1):55-58.

[19] Shafyat A, Asoke K N. Adaptive solution for blind equalization and carrier-phase recovery of square-QAM[J]. IEEE Signal Processing Letters,2010,17(9):791-794.

[20] J Yang, G Dumont. The multimodulus blind equalization and its generalized algorithms[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(5):997-1015.

[21] S Abrar, A Nandi. Blind equalization of square-QAM signals: A multimodulus approach[J]. IEEE Transaction on Communications,2010,58(6):1674-1685.

[22] S Abrar, A Nandi. Adaptive minimum entropy equalization algorithm[J]. IEEE Communications Letters,2010,14(10):966-968.

[23] 李盼痴,王海英,等.量子势阱粒子群优化算法的改进研究[J].物理学报,2012,61(6):1-9.

[24] 郭业才,吴星,黄伟,等.量子人工鱼群优化的自适应最小熵盲均衡算法[J].系统仿真学报,2016,28(2):449-454.

[25] G W Lu, T Miyazaki. Experimental demonstration of RZ-8-APSK generation through optical amplitude and phase multiplexing[J]. IEEE Photonics Technology Letters,2008,20(23):1995-1997.

[26] Z S Liu. APSK constellation with gray mapping[J]. IEEE Communication Letters,2011,15(12):1271-1273.

# 第 7 章　基于 DNA 遗传算法优化的盲均衡算法

**【内容导读】** 本章利用 DNA 遗传算法的优越性能、禁忌搜索的特点和多种群进化原理，对正交小波盲均衡算法进行优化。具体研究内容：基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波盲均衡算法、基于多种群进化禁忌搜索 DNA 遗传算法优化小波分数间隔多模盲均衡算法。

## 7.1　DNA 遗传算法

自 1994 年 Adelman 在 *Science* 杂志上发表首篇关于 DNA 计算的文章以来，DNA 计算迅速成为计算机领域的热点研究内容。由于 DNA 计算的巨大并行性，利用 DNA 计算能够解决数学上复杂的优化组合问题[1-3]。然而，DNA 计算的思想是将所有可能存在的问题解全部枚举出来，需要设计适当的操作步骤将最优解筛选出来，由于受到现代分子生物学水平的限制，

并不能完全实现将最优解从巨大的潜在解空间中筛选出来，因此可能会丢失最优解[4]。为了减少DNA计算的复杂性，很多学者提出了许多改进的DNA计算方法，这些方法主要通过改进DNA编码方式来增加DNA计算的可靠性与准确性。由于目前DNA计算在数学问题上计算精度不高，无法解决工程应用中的复杂问题，因此这种计算方法主要运用于组合优化问题，具有一定的局限性[5-8]。

遗传算法与DNA计算在算法思想上有相似之处，但传统的遗传算法存在搜索效率低、局部搜索能力弱、容易陷入早熟收敛等问题。而DNA是重要的遗传物质，携带着重要的遗传信息，将DNA计算融入遗传算法中，能进一步模拟生物遗传机制、提高遗传算法的搜索性能，就形成了DNA遗传算法。该算法在遗传算法的基础上，采用DNA计算的思想，将种群中的个体编码成腺嘌呤(A)、鸟嘌呤(G)、胞嘧啶(C)和胸腺嘧啶(T)四种碱基组成的碱基序列，并利用基因级的置换交叉操作、转位交叉操作、重构交叉操作等手段对全局最优解进行搜索。

### 7.1.1 DNA遗传算法的主要操作算子

#### 1. 编码与解码方式

(1) DNA编码与解码

DNA遗传算法有两种编码和解码方式。

一种编码方式是受生物体中氨基酸的合成机制的启发而设计出来的，采用三个碱基分子表示变量，即把变量表示成氨基酸合成过程中需要的密码子，通过密码子与氨基酸对照表，就使碱基分子与变量一一对应，不同的氨基酸可以对应[－9,9]之间的整数，然后将这些整数映射到变量对应的变化区间，从而得到与实际问题对应的数值。这种编码方式的优点是编码和解码过程简单，缺点是表示的数据比较少，适用于变量取值为有限离散数据的情况，然而对于大多数自变量连续取值的问题来说，这种编码方式是不适用的。

另一种编码方式是使用一定长度的碱基串来表示问题的一个变量，即把一个碱基串转换为对应的数字，每一个碱基对应一个数字，从而把一个碱基串转换为一个四进制数字串。与 0-1 二进制编码的解码过程类似，首先计算各个基因位上的加权和，然后将所得到的整数映射到该自变量的变化区间，从而得到该自变量的对应实数码。这种编码的精度由每个自变量对应的碱基串的长度决定，在相同的编码长度时这种编码精度高于二进制编码的精度，通过这种编码方式可以引入复杂的基因遗传操作，因而更适合用于求解优化问题。

由于盲均衡算法的代价函数是连续函数，因此本书采用第二种编码方式。首先将 DNA 分子抽象为由腺嘌呤(A)、鸟嘌呤(G)、胞嘧啶(C)和胸腺嘧啶(T)四种碱基组成的碱基序列。为了便于计算机处理，采用 0、1、2、3 这四个数字分别对应四种 DNA 碱基，其编码空间为 $E=\{0,1,2,3\}^l$。在多种关系映射中，采用的映射方式为：0123/CGAT，通过这种映射关系，可以将一串 DNA 碱基序列转换为数字序列，同时碱基的数字编码也要体现互补碱基对之间的配对规律。

对于 $n$ 维最小优化问题可以表述为

$$\begin{cases}\min J(x_1, x_2, \cdots, x_n) \\ x_{i\min} \leqslant x_i \leqslant x_{i\max}, \quad i = 1,2,\cdots,n\end{cases} \tag{7.1.1}$$

式中，$x_i(i=1,2,\cdots,n)$代表控制变量，可以表示成长度为 $l$ 的四进制数字串；$J(x_1, x_2, \cdots, x_n)$是目标函数；$x_{i\min}$和 $x_{i\max}$分别为每个变量对应的最小值与最大值，每个变量的编码精度为$(x_{i\max}-x_{i\min})/4^l$。

DNA 遗传算法的解码方式与二进制解码方式相似，首先将四进制数解码成十进制数，即

$$\text{dec } x_i = \sum_{j=1}^{l} \text{bit}(j) \times 4^{l-j} \tag{7.1.2}$$

式中，bit($j$)是四进制数据的位数字。第二步根据变量的不同取值范围转换为对应问题的解，即

$$x_i = \frac{\text{dec } x_i}{4^{l-1}}(x_{i\max} - x_{i\min}) + x_{i\min} \tag{7.1.3}$$

这种 DNA 碱基编码方式可以将更多的基因级操作引入到 DNA 遗传算法中，从而提高算法的搜索性能[9]。

(2) DNA 编码约束条件[5,7,9]

DNA 计算是通过 DNA 序列的切割、删除等操作来实现的，但其在操作过程中，需要严格的编码约束条件来制约 DNA 编码序列，从而保持序列的多样性，提高编码质量。现介绍两个编码约束条件。

① 汉明距离约束

两个 DNA 序列的汉明距离是所有对应位置字符不同的总数。设 DNA 序列 $X$ 和 $Y$ 分别为 $X=5'-x_1x_2\cdots x_n-3'$和 $Y=5'-y_1y_2\cdots y_n-3'$，其汉明距离记为 $H(X,Y)$且计算公式为

$$H(X,Y)=\sum_{i=1}^{n}h(x_i,y_i),\quad h(x_i,y_i)=\begin{cases}0, & x_i=y_i\\ 1, & x_i\neq y_i\end{cases}\tag{7.1.4}$$

在 DNA 遗传算法中，若两个 DNA 序列太过相似，则有可能导致算法陷入局部极值的问题。汉明距离越大，说明两个 DNA 序列 $X$ 和 $Y$ 之间不同的碱基个数就越多，因此，发生特异性杂交的可能性就越小。

汉明补距离 $H_1(X,X)$：从生物学角度看，DNA 分子处于三维空间中，容易与自身发生杂交。为避免这种杂交的发生，提出汉明补距离。其定义为

$$H_1(X,X)=\min_{-n<k<n}H(X,\sigma^k(X^{\mathrm{R}}))\tag{7.1.5}$$

式中，当 $k>0$ 时，$\sigma^k$ 表示 $X^{\mathrm{R}}$ 序列右移；当 $k<0$ 时，$\sigma^k$ 表示 $X^{\mathrm{R}}$ 序列左移。$k$ 表示移动的位数；$X^{\mathrm{R}}$ 表示 DNA 序列 $X$ 的反链。当 $H_1$ 值较小时，$X$ 容易和自身的反链相互杂交；当 $H_1$ 值很大时，$X$ 和 $X^{\mathrm{R}}$ 几乎不存在互补的 DNA 碱基，因此不会与自身产生杂交。

② 相似度约束

相似度约束是指 DNA 序列 $X$ 和 $Y$ 中碱基的相似度，而相似度可以通过计算序列 $X$ 和 $Y$ 之间移动后取最小汉明距离得到，其计算公式为

$$H_2(X,Y)=\min_{-n<k<n}H(X,\sigma^k(Y))\tag{7.1.6}$$

式中，当 $k>0$ 时，$\sigma^k$ 表示 $Y$ 序列右移；当 $k<0$ 时，$\sigma^k$ 表示 $Y$ 序列左移。当 $H_2$ 值较小时，序列 $X$ 和 $Y$ 非常相似，容易出现非特异性杂交；当 $H_2$ 值很大时，$X$ 和 $Y$ 相同碱基很少，很难出现非特异性杂交现象。

2. 交叉操作

交叉操作是 DNA 遗传算法的一个重要组成部分，是种群维持多样性的重要保证，是模拟自然界生物体有性繁殖而设计的操作算子。本节采用 DNA 碱基的编码方式更加有利于设计各种基因级的交叉操作算子。根据现代分子生物学基因操作和 DNA 碱基编码方式设计了三种交叉操作算子：置换交叉算子、转位交叉算子和重构交叉算子[11~13]。根据种群个体的适应度值将种群分为优质种群和劣质种群，在不同的种群中分别进行不同的交叉操作，增加了种群的多样性，使种群朝着最优解方向进化。交叉操作是根据 DNA 转座子的思想设计出来的。DNA 转座子是一段可以移动的 DNA 序列，DNA 转座是将一段 DNA 序列转移到另一个位置上，在这个过程中，可以使两个原本相距较远的 DNA 序列组合到一起，从而产生新的 DNA 序列。基因转座过程如图 7.1 所示。

① 置换交叉操作

随机选取(0,1)之间的数作为执行概率，若执行概率比置换交叉概率 $p_1$ 小，则在 DNA 种群中随机选取两个 DNA 序列作为父体，在置换交叉操作过程中，从两个父体 DNA 序列中随机选取一段相同长度的碱基序列，并且交换两个碱基序列的位置，得到新的 DNA 序列。置换交叉操作如图 7.2 所示。

② 转位交叉操作

随机选取(0,1)之间的数作为执行概率，若执行概率比转位交叉概率 $p_2$ 小，则在 DNA 种群中随机选取一个 DNA 序列作为父体，并在该序列中随机选取一段碱基序列插入到该序列的随机位置中，从而产生新的 DNA 序列。该操作如图 7.3 所示。

转座子 靶位
供体 受体
转座子末端切开 靶位点切开
形成中间体
转座子末端切开
中间体解开
转座子复制
中间体解开
受体 受体
供体 供体

图 7.1 基因转座过程

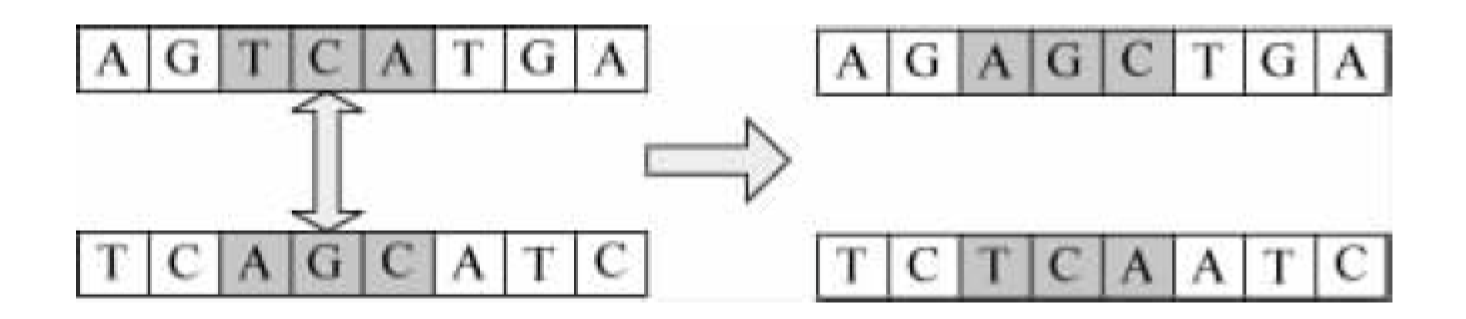



图 7.2 置换交叉操作

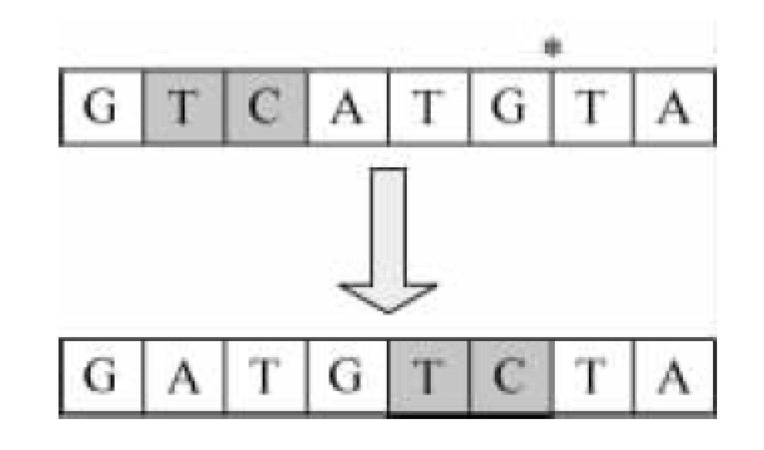



图 7.3 转位交叉操作

③ 重构交叉操作

对于 DNA 遗传算法来说，种群个体适应度值越大的个体进入下一代的概率越大。当种群中出现适应度值很大的个体时，有时候种群中会出现大部分个体的适应度值很接近，从而导致种群多样性下降，此时普通的交叉操作很难再提高种群的多样性，尽管变异操作能在一定程度上改善这种情况，但由于变异概率比较低，因此种群的多样性下降问题依然不能得到有效改善，最终导致局部最优的出现。为了克服此问题，可采用重构交叉算子，该算子通过重构种群中相似度高的个体，并保留原个体的优秀基因，从而使种群具有多样性。

由于重构交叉操作是为提高种群个体的多样性而设计的，因此选取用于重构交叉操作的父体需要遵循一定的原则。首先在优质种群中选择一个用于重构交叉的父体；其次，在该优质种群中随机选择两个个体作为备选个体；再次，用已知父体分别与两个备选父体分别比较相似度，选择与已知父体相似度大(适应度值差值较小)的个体作为另一个父体用于重构交叉操作，分别标记两个父体为父体 A 和父体 B。

在父体 A 的末端剪切一段序列粘贴到父体 B 的首部，为了保持个体碱基序列长度保持不变，将父体 B 尾部多余的碱基序列切除，同时随机生成一段与被切除序列等长度的碱基片段粘贴到父体 A 的首部。完成操作后，生成两个序列长度相等的子代个体。重构交叉操作，如图 7.4 所示。

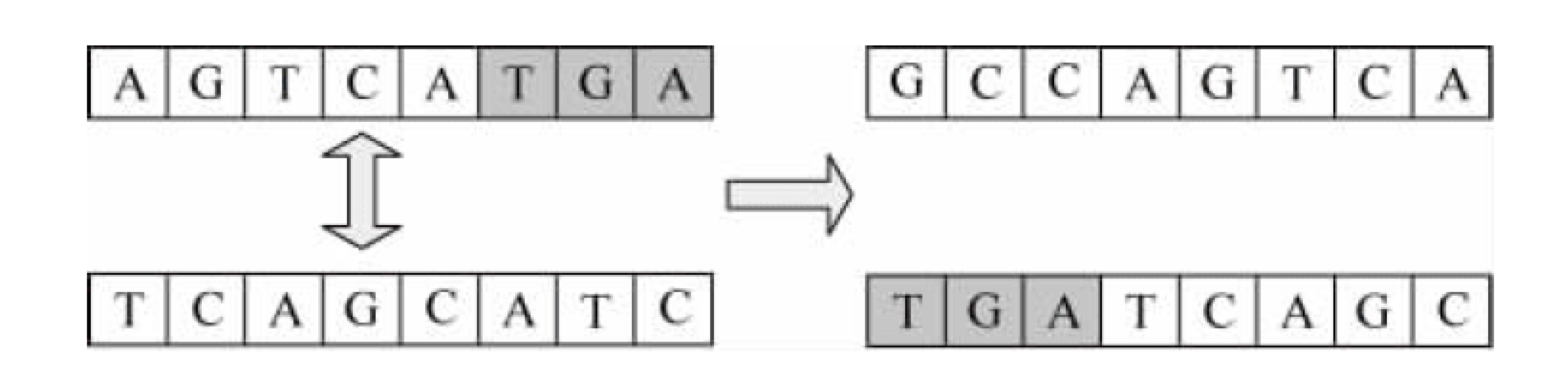



图 7.4　重构交叉操作

3. 变异操作

变异操作是 DNA 遗传算法的重要组成部分，是模拟生物体基因突变现象而设计的操作算子，目的在于提高种群的多样性。传统的变异操作是

单点变异,即在基因序列中随机选择一个变异位点以固定的变异概率进行变异操作。这种变异操作变异位点少,变异概率不能调节,难以使种群保持多样性,容易导致早熟收敛。

(1) 基本变异操作

随机选取(0,1)之间的数作为执行概率,若执行概率比变异概率 $p$ 小,则在 DNA 种群中随机选取一个 DNA 序列作为父体,将该父体中出现频率最高的碱基用父体中出现频率最低的碱基代替,从而产生新的 DNA 序列,如图 7.5 所示。

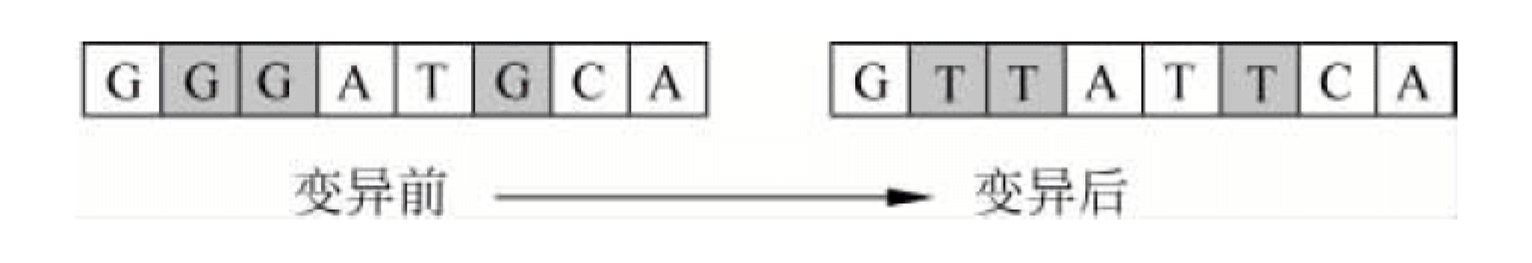



图 7.5 变异操作

(2) 自适应变异操作

自适应变异操作的每一代的变异概率随着进化的代数变化而变化。根据生物学原理,在 DNA 序列中存在着“热点”和“冷点”区域,“热点”区域的变异概率高于“冷点”区域的变异概率。基于这一生物原理,将 DNA 序列分为高位和低位部分,在进化初始阶段,高位部分具有较高的变异概率,以加快收敛速度;在进化后期阶段,要求 DNA 序列的高位部分具有较小的变异概率,低位部分具有较大的变异概率,以实现对最优解的精确搜索。高位和低位的变异概率公式为[13]

$$p_{mh} = a_1 + \frac{b_1}{1 + \exp[a(g - g_0)]} \tag{7.1.7}$$

$$p_{ml} = a_1 + \frac{b_1}{1 + \exp[-a(g - g_0)]} \tag{7.1.8}$$

式中,$p_{mh}$ 和 $p_{ml}$ 分别代表高位部分和低位部分的变异概率,$a_1$ 表示初始时刻的变异概率值,$a_1=0.02$;$b_1$ 表示变异概率的变化值,$b_1=0.2$;$g$ 表示当前的进化代数,$g_0$ 表示变异概率变化最大时的进化代数值;$a$ 是变异概率最大时的斜率,$a=0.2$。变异概率曲线如图 7.6 所示。

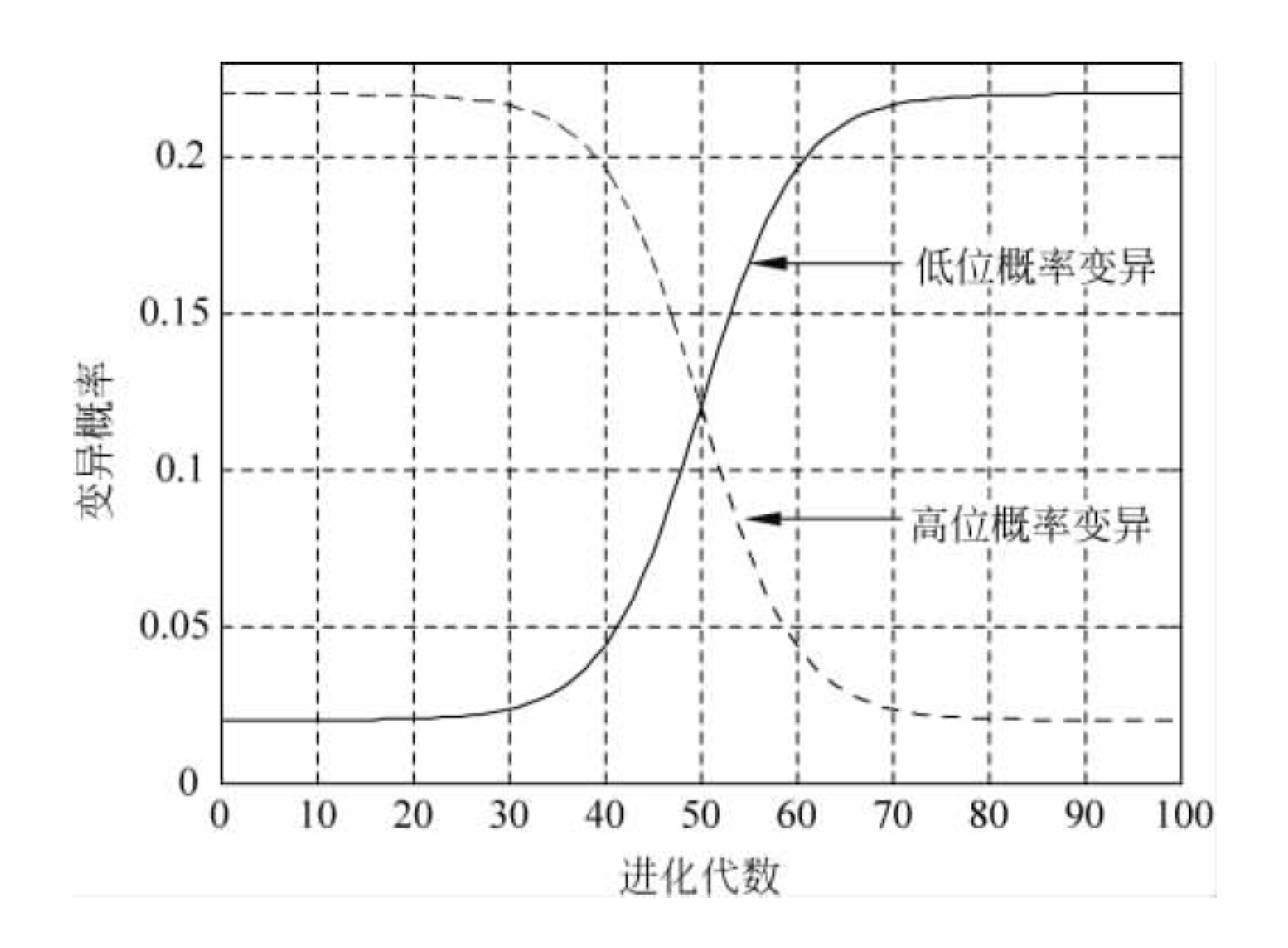



图 7.6 DNA 碱基变异概率

4. 选择操作

本节使用联赛选择方法产生新种群。联赛选择就是在种群中随机选择一定量的个体,挑选出适应度值最大的个体进入下一代,再将剩余的个体放回原种群,重复以上过程,直到子代个体数量达到要求值。另外,为了防止将种群中适应度值最大的个体丢失掉,本节采用精英保留机制,即直接把父代种群中适应度值最大的个体复制到下一代。

### 7.1.2 DNA 遗传算法操作

根据 DNA 编码规则及交叉、变异和选择操作,将 DNA 遗传算法的操作步骤归纳如下。

步骤 1:首先设置算法的相关参数。设置最大进化代数 $G$,种群规模 $M$,DNA 序列编码长度 $L$,算法终止阈值 $\Delta$,置换交叉概率 $p_1$、转位交叉概率 $p_2$ 和重构交叉概率 $p_3$。

步骤 2:确定适应度函数并计算适应度值。初始化种群,随机生成 $M$ 个长度为 $N\times L$ 的 DNA 序列构成初始种群。根据 7.1.1 节的编码规则将每个 DNA 序列解码,然后代入到适应度函数中计算每个 DNA 个体的适应

度值。

步骤 3：种群分组。将整个种群作为搜索空间，根据个体适应度值的大小将所有个体进行排序，前一半个体为优质种群，后一半个体为劣质种群，并且将种群中个体适应度值最大的个体作为精英个体保留。

步骤 4：执行交叉操作。交叉操作主要是在优质种群中完成的，首先在优质种群中随机选择两个个体作为父体，然后对被选中的父体分别以概率 $p_1$ 和 $p_2$ 执行置换交叉操作和转位交叉操作；如果置换交叉操作和转位交叉操作均未被执行，则按概率 $p_3$ 执行重构交叉操作，每次交叉操作产生的新个体不放回原种群，重复以上交叉操作直到产生 $M/2$ 个新个体，最后将产生的新个体和劣质种群一起放入到优质种群中，从而形成种群规模为 $3M/2$ 个个体的混合种群。

步骤 5：对种群执行变异操作。对混合种群中的个体执行变异操作，变异操作采用自适应动态变异操作。对于种群中的每一个个体，都执行一次变异操作，然后将变异后产生的新个体取代原个体。变异操作完成后，重复执行 $M-1$ 次联赛选择操作，选择出 $M-1$ 个个体，与原来的精英个体一起组成新种群，最后计算种群个体适应度值，将适应度值最大的个体作为最优个体。种群进化代数加 1。

步骤 6：判断进化条件。如果达到设置的最大进化代数 $G$ 或者当前最优解的个体适应度值变化小于阈值 $\Delta$，则将种群中适应度值最大的个体作为最优个体输出，解码后的值作问题的最优解；否则，返回步骤 2。

DNA 遗传算法流程如图 7.7 所示。

图 7.7 DNA 遗传算法的流程

# 7.2 基于 DNA 遗传算法优化的正交小波常模盲均衡算法

## 7.2.1 算法原理

现利用 DNA 遗传算法的全局搜索能力强、搜索效率高、局部搜索性能好、不依赖于具体问题领域等优点，对正交小波常模盲均衡算法（WTCMA）进行优化，得到的基于 DNA 遗传算法优化的正交小波常模盲均衡算法（DNA Genetic Algorithm based WTCMA，DNAGA-WTCMA）[14-15]，如图 7.8 所示。

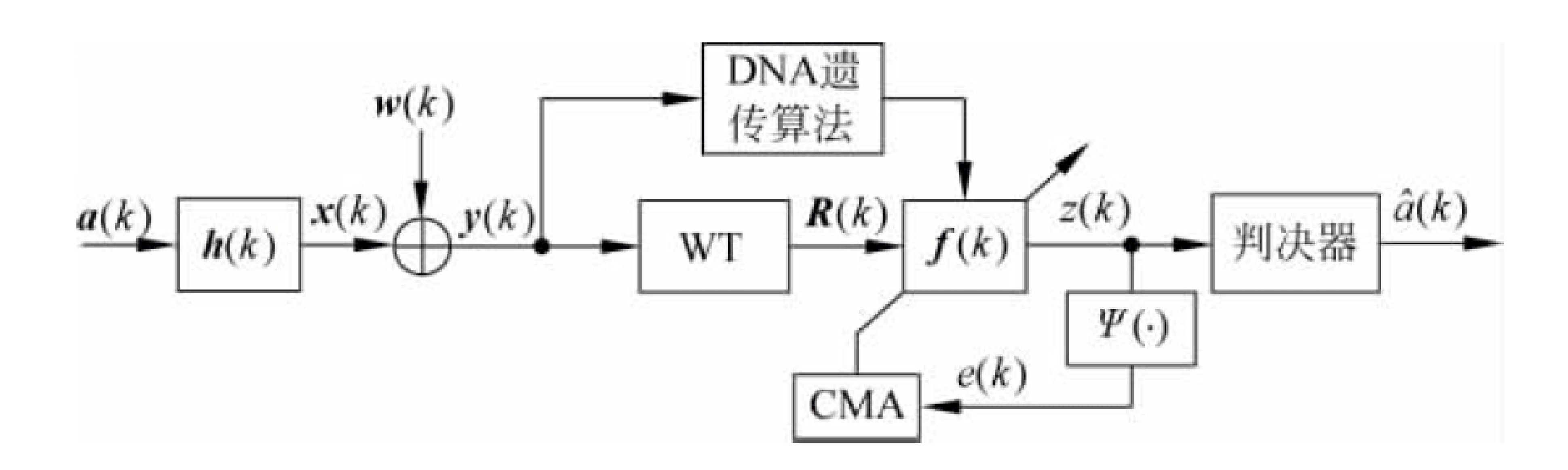



图 7.8 DNAGA-WTCMA 原理

图中，$\boldsymbol{a}(k)$是信号源发射的原始信号；$\boldsymbol{h}(k)$是未知信道的冲激响应；$\boldsymbol{x}(k)$是信道输出信号；$\boldsymbol{w}(k)$是信道输出端的加性高斯白噪声；$\boldsymbol{y}(k)$是均衡器的接收信号；$\boldsymbol{R}(k)$是 $\boldsymbol{y}(k)$经正交小波变换后的信号或均衡器输入信号；$\boldsymbol{f}(k)$是均衡器权向量；$z(k)$是均衡器输出信号；$\Psi(\cdot)$是误差生成函数，$e(k)$是误差信号；$\hat{a}(k)$是判决装置对 $z(k)$的判决输出信号。

均衡器的输入信号为

$$\boldsymbol{y}(k)=\boldsymbol{h}^{\mathrm{T}}\boldsymbol{a}(k)+\boldsymbol{w}(k) \tag{7.2.1}$$

经过小波变换后信号为

$$\boldsymbol{R}(k)=\boldsymbol{V}\boldsymbol{y}(k) \tag{7.2.2}$$

均衡器的输出信号为

$$z(k)=\boldsymbol{f}^{\mathrm{H}}(k)\boldsymbol{R}(k) \tag{7.2.3}$$

误差信号为

$$e(k) = R^2 - |z(k)|^2 \tag{7.2.4}$$

式中，$\boldsymbol{V}$ 为正交小波变换矩阵，上标 H 表示共轭转置。

WTCMA 的代价函数为

$$J(k) = E\{[R^2 - |z(k)|^2]^2\} \tag{7.2.5}$$

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + \mu \hat{\boldsymbol{R}}^{-1}(k)\boldsymbol{R}(k)e^*(k)z^*(k) \tag{7.2.6}$$

式中，$\mu$ 为步长，$\hat{\boldsymbol{R}}(k)=\mathrm{diag}[\sigma_{j,0}^2(k),\sigma_{j,1}^2(k),\cdots,\sigma_{j,k_j}^2(k),\sigma_{J+1,0}^2(k),\cdots,\sigma_{J+1,k_J}^2(k)]$，其中 diag[]表示对角矩阵，$\sigma_{j,k_j}^2(k)$和 $\sigma_{J+1,k_J}^2(k)$分别表示 $u_{j,n}(k)$和 $s_{J,n}(k)$的平均功率估计，其递推估计公式为

$$\begin{cases}\hat{\sigma}_{j,n}^2(k+1) = \beta\hat{\sigma}_{j,n}^2(k) + (1-\beta)\,|u_{j,n}(k)|^2 \\ \hat{\sigma}_{J+1,n}^2(k+1) = \beta\hat{\sigma}_{J,n}^2(k) + (1-\beta)\,|s_{J,n}(k)|^2\end{cases} \tag{7.2.7}$$

式中，$\beta$ 为平滑因子，且 $0<\beta<1$，一般取略小于 1 的数，$u_{j,n}(k)$和 $s_{J,n}(k)$为信号经尺度函数 $\varphi_{j,n}(k)$和小波函数 $\phi_{j,n}(k)$卷积生成的变换系数，即

$$\begin{cases}u_{j,n}(k) = \sum_{i=0}^{N-1} x(k-i)\varphi_{j,n}(i) \\ s_{J,n}(k) = \sum_{i=0}^{N-1} x(k-i)\phi_{J,n}(i)\end{cases} \tag{7.2.8}$$

以上就是正交小波常模盲均衡算法（WTCMA）。然而，这种算法容易陷入局部最优。现由 DNA 遗传算法优化 WTCMA 的操作步骤如下。

步骤 1：初始化种群。设置最大进化代数 $G_{\max}$或者阈值 $\Delta$。设 DNA 遗传算法的初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$，其中 $\boldsymbol{f}_m$ 对应于 WTCMA 的一个权向量，$1\leqslant m\leqslant M$，$M$ 为种群规模中个体数量；采用 A、G、C、T 四种碱基对盲均衡算法的权向量进行编码，编码空间为 $E=\{A,G,C,T\}^L$，其中 $L=L_f\times l$ 为每个个体 DNA 序列的长度，$L_f$ 为盲均衡器权长，$l$ 表示用 DNA 编码均衡器权向量中的每一个抽头系数所需要的碱基数。

步骤 2：确定适应度函数。

将 WTCMA 代价函数的倒数作为 DNA 遗传算法的适应度函数，即

$$\mathrm{Fit}(\boldsymbol{f}_m)=\frac{b}{J(\boldsymbol{f}_m)} \tag{7.2.9}$$

式中,$b$ 表示比例系数。

步骤 3:对种群分组。以整个种群作为搜索空间,将均衡器的接收信号作为 DNA 遗传算法的输入信号,计算种群中每个个体的适应度值,根据个体适应度值大小将所有个体进行排序,前一半个体为优质种群,后一半个体为劣质种群,并且将种群中个体适应度值最大的个体作为精英个体保留。

步骤 4:执行交叉操作。首先在优质种群中随机选择两个个体作为父体,然后对被选中的父体分别以概率 $p_1$ 和 $p_2$ 执行置换交叉操作和转位交叉操作;如果置换交叉操作和转位交叉操作均未被执行,则按概率 $p_3$ 执行重构交叉操作,每次交叉操作产生的个体不放回原种群,重复以上交叉操作直到产生 $M/2$ 个新个体,最后将产生的新个体和劣质种群一起放入到优质种群中,从而形成种群规模为 $3M/2$ 个个体的混合种群。

步骤 5:混合种群的变异、选择操作。对混合种群中的个体执行变异操作,变异操作采用自适应动态变异操作。对于种群中的每一个个体,都执行一次变异操作,然后将变异后产生的个体取代原个体。变异操作完成后,重复执行 $M-1$ 次联赛选择操作,选出 $M-1$ 个个体,与原来的精英个体一起组成新种群,最后计算每个个体的适应度值,将适应度值最大的个体作为最优个体。种群进化代数加 1。

步骤 6:判断是否达到进化终止条件。如果达到设置的最大进化代数 $G_{\max}$ 或者当前最优解的个体适应度值变化小于阈值 $\Delta$,则将种群中适应度值最大的个体作为最优个体输出,解码后的值作均衡器权向量的最优值;否则,返回步骤 3。

### 7.2.2 算法仿真

为了验证 DNAGA-WTCMA 的有效性,以 CMA 和 WTCMA 为比较对象,进行仿真实验。

**【实验 7.1】** 信道 $\boldsymbol{h}$=[0.3132 −0.1040 0.8908 0.3134],信道噪声为高斯白噪声,发射信号为 8PSK 信号,均衡器权长为 16,步长 $\mu_{\text{CMA}}=0.002$,$\mu_{\text{WTCMA}}=0.0023$,$\mu_{\text{DNAGA-WTCMA}}=0.003$,信噪比为 20dB,训练样本个数 $N=8000$。对信道的输入信号采用 DB2 正交小波分解,功率初始值设置为 4,遗忘因子 $\beta=0.99$。DNAGA-WTCMA 种群规模取 30,置换交叉概率 $p_1=0.8$,转位交叉概率 $p_2=0.5$,重构交叉概率 $p_3=0.2$。变异操作按自适应变异概率执行。终止进化代数为 100 代。

当信道 $\boldsymbol{h}$=[0.9656 −0.0906 0.0578 0.2368]时,信道噪声为高斯白噪声,发射信号为 16QAM,均衡器权长为 16,步长 $\mu_{\text{CMA}}=0.0001$,$\mu_{\text{WTCMA}}=0.0002$,$\mu_{\text{DNAGA-WTCMA}}=0.000\,25$,信噪比为 20dB,训练样本个数 $N=8000$。对信道的输入信号采用 DB2 正交小波分解,功率初始值设置为 4,遗忘因子 $\beta=0.99$。DNAGA-WTCMA 种群规模取 50,终止进化代数为 100 代。

在保证眼图完全清晰睁开的前提下,500 次蒙特卡罗仿真结果如图 7.9 所示。

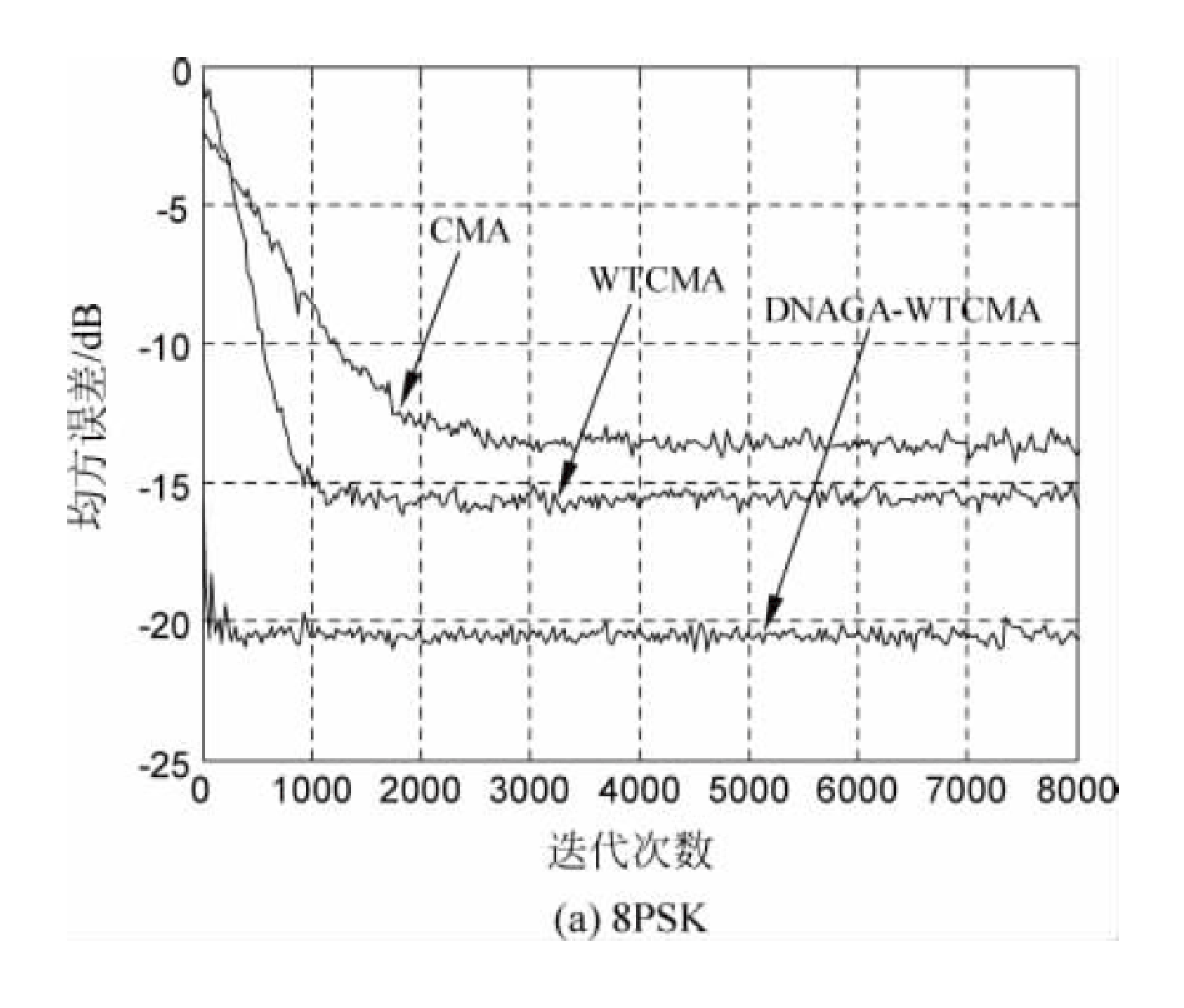
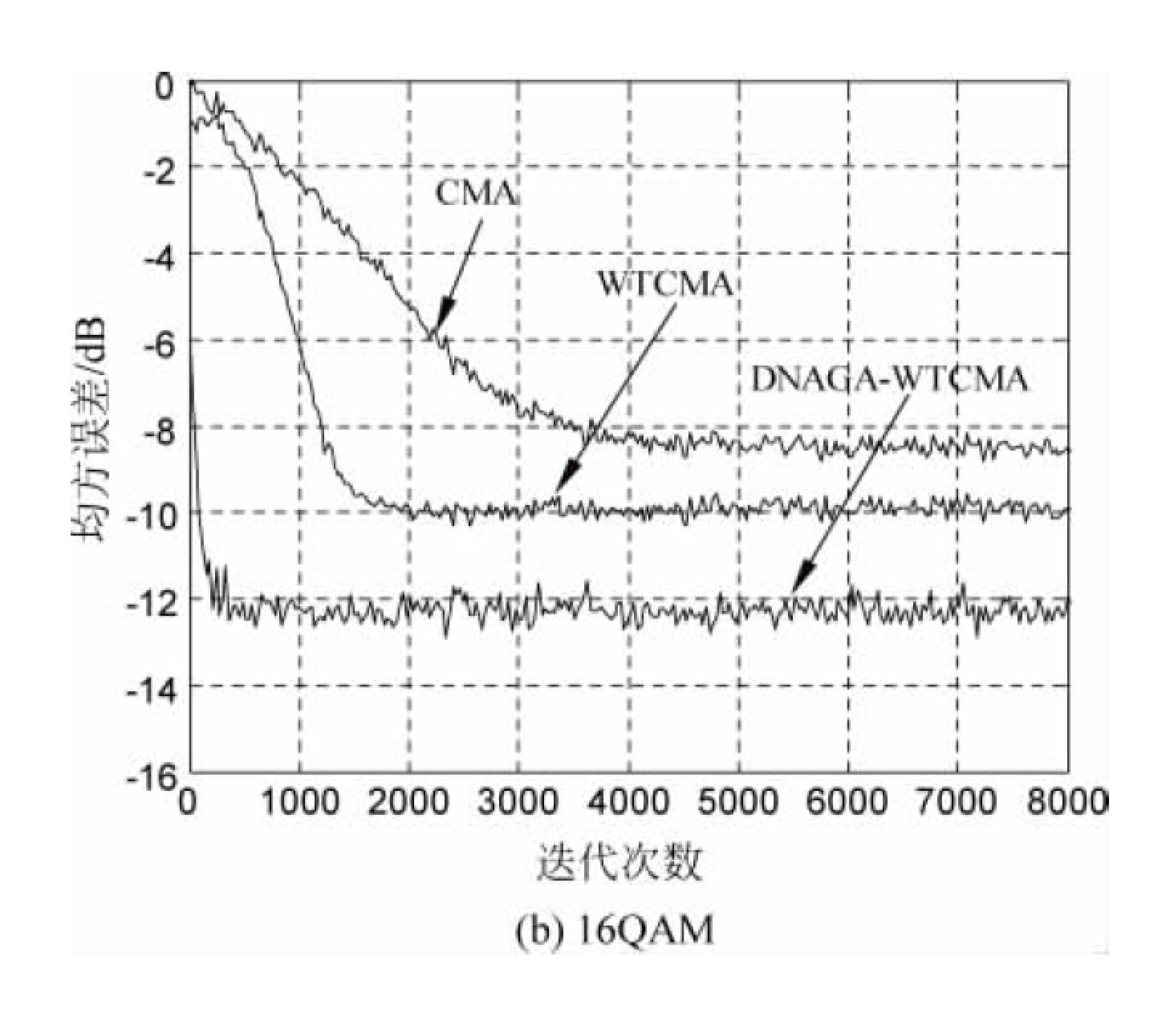



图 7.9 仿真结果



(b) 16QAM

图 7.9 （续）

图 7.9(a)表明,DNAGA-WTCMA 的收敛速度比 WTCMA 快约 1100 步,比 CMA 快约 3000 步；在稳态误差上,DNAGA-WTCMA 的稳态误差最低。图 7.9(b)表明,DNAGA-WTCMA 的收敛速度比 WTCMA 快约 1800 步,比 CMA 快约 3800 步；在稳态误差上,DNAGA-WTCMA 的稳态误差最低。

## 7.3 基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波常模盲均衡算法

盲均衡算法是利用随机梯度搜索算法对均衡器权向量寻找最优解的,这种方法要求盲均衡算法的代价函数连续可导,并且容易陷入局部最优。利用 DNA 遗传算法对均衡器权向量进行最优化,能够显著提高盲均衡算法的性能。然而,在一般的 DNA 遗传算法中,种群的交叉概率是确定值,不能随种群变化而做适当的调整,搜索效率不高。另外,DNA 遗传算法搜索问题最优解时,有时会出现重复搜索,影响搜索效率。针对这些缺陷,现结合禁忌搜索算法,同时采用 DNA 遗传算法中的交叉操作概率随种群进

化代数而变化的特性，研究了一种基于禁忌搜索自适应双链 DNA 遗传算法优化的常模盲均衡算法。

### 7.3.1 禁忌搜索算法

禁忌搜索（Tabu Search 或 Taboo Search，TS）算法[16-23]是由 F. Glover 于 1986 年提出的一种亚启发式搜索算法，是对局部领域搜索的拓展，是一种全局逐步寻优算法。它能够采用一种灵活的“记忆”技术记录已经搜索过的路径，并且根据记录的知识再选择合适的搜索路径，提高了搜索效率、避免了局部收敛。为了克服搜索过程中可能出现的重复搜索，禁忌搜索算法在搜索过程中设置了一个灵活的禁忌表，对已经搜索过的路径和相关的信息进行记录，在下一次搜索的过程中，会根据禁忌表中记录的内容对搜索路径进行选择，指明搜索方向，避免对已经搜索过的路径重复搜索，保证搜索路径的多样性。由于 TS 算法独特的运行机制及较高的搜索效率等优点，已经被广泛应用于参数选择、生产调度和神经网络等领域，并取得了很大的成功。

禁忌搜索算法涉及邻域（neighborhood）、候选解（candidate）、禁忌长度（tabu length）、藐视准则（aspiration criterion）等概念。

TS 算法流程如图 7.10 所示。具体步骤如下。

步骤 1：设置算法参数和初始解 $x(0)$、当前解 $x_{now}=x(0)$、当前最优解 $x_{best}=x(0)$以及禁忌表 Table_list$=\phi$。

步骤 2：如果算法满足终止条件，则停止搜索，输出结果；否则，继续步骤 3。

步骤 3：产生当前解的邻域 $N(x_{now})$，从中选取若干解作为候选解集 Can_N($x_{now}$)。

步骤 4：判断是否有候选解满足藐视准则。若存在，则用满足藐视准则的最佳候选解 $x_{opt}$ 替代 $x_{now}$ 成为新的当前解，即 $x_{now}=x_{opt}$，并替换当前最优解，即 $x_{best}=x_{opt}$，同时更新禁忌表 Table_list（如果禁忌表中禁忌对象采用

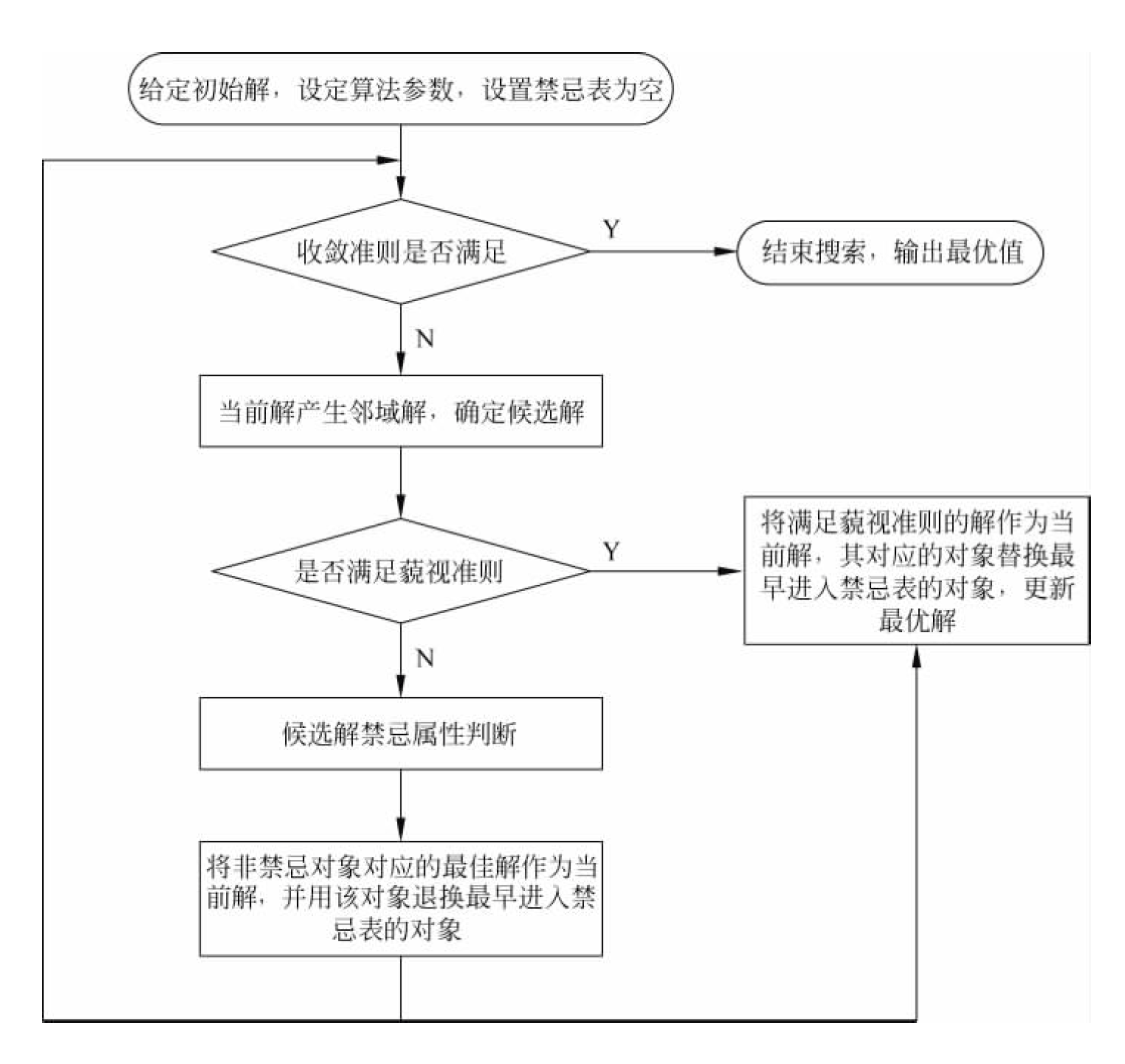



图 7.10 禁忌搜索算法流程

先进先出方式，则用与 $x_{best}$ 对应的禁忌对象替换最早进入禁忌表的禁忌对象)，然后转到步骤 2；否则，继续步骤 5。

步骤 5：判断各候选解对应对象的禁忌属性，选择候选解集 Can_N($x_{now}$)中非禁忌的最佳候选解 $x_{non\text{-}opt}$ 替代 $x_{now}$ 成为新的当前解，即 $x_{now}=x_{non\text{-}opt}$，同时更新禁忌表 Table_list，然后转步骤 2。

可见，邻域函数、禁忌对象、禁忌表以及藐视准则构成了 TS 算法的关键。其中，邻域函数主要采用局部邻域搜索的思想，用于实现邻域搜索；设置禁忌表和禁忌对象目的是为了避免迂回搜索，从而跳出局部最优；藐视准则的设置是为了避免丢失优良的状态，激励对优良状态的局部搜索，进而

实现全局最优。需要指出的是，上述 TS 算法仅是一般的算法流程，对算法中的关键步骤以及禁忌对象的改变可以设计出不同的禁忌搜索算法。

### 7.3.2 基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波常模盲均衡算法

将 TS 算法和 DNAGA 相结合，并采用双链模型，优化常模盲均衡算法，就形成了基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波常模盲均衡算法（TS algorithm and DNAGA based WTCMA，TSDNAGA-WTCMA）[14,24]。

#### 1. DNA 双链模型

DNA 是储存生物遗传信息的重要物质，在细胞核内以双链反螺旋结构的形式存在。为了进一步提高 DNA 遗传算法的性能，本节采用双链 DNA 遗传算法，将种群中的每一个个体使用一个具有互补双链的分子序列来表示，并将其应用到盲均衡算法中，从而进一步提高盲均衡算法的性能。

#### 2. 禁忌搜索交叉操作

将 TS 算法的思想应用到 DNA 遗传算法的交叉算子中，就形成了禁忌交叉算子。禁忌交叉算子使用禁忌表存储种群中的个体适应度。用渴望水平取父代群体的平均适应度值。在禁忌交叉操作时，如果子代的适应度值比渴望水平大，则渴望水平取子代适应度值，子代适应度值放入禁忌表中；否则，如果子代个体的适应度值在禁忌表中，则取父体中适应度值最大的个体进入下一代，如果子代个体的适应度值不在禁忌表中，则将子代取代父代进入下一代，并将子代适应度值放入禁忌表中。从以上操作过程可以看出，适应度值较大的个体进入下一代的机会是很大的，但由于使用了禁忌表，限制了个体适应度相似度高的个体在子代出现的次数，因此不是所有的适应度值大的个体都能进入到下一代，从而能够保持种群中个体的多样性，避免

算法出现早熟收敛。禁忌交叉操作流程伪码为($S_{best_so_far}$表示渴望水平):

```
Begin
    父体 S_k^1, S_k^2
    S_best_so_far = 种群平均适应度
If Fitness(S_k^1)> Fitness(S_k^2)
    S_k^max = S_k^1;
Else
    {S_k^max = S_k^2;
    父体 S_k^1, S_k^2 经过置换、转位、重构交叉后生成子代 S_{k+1}^1, S_{k+1}^2;
    T(S_k^1, S_{k+1}^1, S_k^max); T(S_k^2, S_{k+1}^2, S_k^max);}
End
Function T(S_k^1, S_{k+1}^1, S_k^max)
If Fitness(S_{k+1}^1)> S_best_so_far
    {子代 S_{k+1}^1 取代父体 S_k^1;
    Fitness(S_{k+1}^1)加入禁忌表;
    S_best_so_far = Fitness(S_{k+1}^1);}
Else if Fitness(S_{k+1}^1)在禁忌表中
        取父体中适应值最大的个体进入下一代;
    Else{子代 S_{k+1}^1 取代父体 S_k^1;
    Fitness(S_{k+1}^1)放入禁忌表中;}
    End
End
T(S_k^2, S_{k+1}^2, S_k^max)与 T(S_k^1, S_{k+1}^1, S_k^max)类似。
```

在禁忌搜索交叉操作中包含7.1.1节三种类型的交叉算子,即置换交叉算子、转位交叉算子和重构交叉算子。

### 3. 自适应交叉概率

早熟收敛问题是影响DNA遗传算法搜索性能重要因素之一,早熟收敛往往会使种群的多样性下降、陷入局部最优。在DNA遗传算法中,交叉概率的大小决定了种群进化的快慢,交叉概率过大或过小都会对算法的搜索性能造成一定的影响,因此,合理的选择交叉概率是提高DNA遗传算法的关键。传统的DNA遗传算法采用固定的交叉概率对种群个体进行交叉

操作，在算法搜索后期，可能因为交叉概率过大而不能快速搜索到最优解，影响搜索速度。为了解决这一问题，本节采用自适应交叉概率，在种群进化过程中，交叉概率将随着种群进化的程度自适应调整。

首先设置 $\alpha$ 分位适应度函数。设第 $k$ 代种群中第 $i$ 个个体的适应度值为 $\mathrm{Fit}_k^i$，$1<i<N$，$N$ 为种群规模。$\mathrm{Fit}_{k_\max}$ 表示种群中第 $k$ 代最优个体的适应度值，$\overline{\mathrm{Fit}}_k$ 为种群的适应度平均值，$\overline{\mathrm{Fit}}_{k_\mathrm{big}}$ 表示大于适应度平均值的所有个体的适应度平均值，$\alpha$ 分位适应度值函数 $\hat{\mathrm{Fit}}_k^\alpha$ 定义为

$$\hat{\mathrm{Fit}}_k^\alpha=\begin{cases}\mathrm{Fit}_k^{(\lceil N_\alpha\rceil+1)}, & N_\alpha\notin Z\\ \dfrac{(\mathrm{Fit}_k^{N_\alpha}+\mathrm{Fit}_k^{(N_\alpha+1)})}{2} & N_\alpha\in Z\end{cases} \tag{7.3.1}$$

式中，$0<\alpha<1$，$N_\alpha=\alpha N$，$\lceil\ \rceil$ 表示向上取整函数。

三均值适应度定义为

$$\overline{\mathrm{Fit}}_{k_\mathrm{ave}}^{3'}=\frac{1}{4}\hat{\mathrm{Fit}}_k^{0.25}+\frac{1}{2}\hat{\mathrm{Fit}}_k^{0.5}+\frac{1}{4}\hat{\mathrm{Fit}}_k^{0.75} \tag{7.3.2}$$

早熟程度指标 $P_r$ 定义为

$$P_r=(\mathrm{Fit}_{k_\max}-\overline{\mathrm{Fit}}_{k_\mathrm{big}}^{3'})/\mathrm{Fit}_{k_\max} \tag{7.3.3}$$

式中，$\overline{\mathrm{Fit}}_{k_\mathrm{big}}^{3'}$ 为大于三均值适应度值的所有个体的适应度平均值。

遗传操作的执行概率根据早熟程度指标 $p_r$ 设置，基本交叉概率定义为

$$p_c=(1+\exp(-\tau_c p_r))^{-1},\quad \tau_c>0 \tag{7.3.4}$$

式中，由基本交叉概率可得相应的置换交叉、转位交叉、重构交叉操作的执行概率分别为

$$p_1=0.75p_c,\quad p_2=0.5p_c,\quad p_3=0.25p_c \tag{7.3.5}$$

#### 4. 禁忌搜索自适应双链 DNA 遗传算法优化流程

禁忌搜索自适应双链 DNA 遗传算法优化正交小波常模盲均衡算法的操作步骤如下。

步骤 1：首先按 7.1.1 节的编码方式将均衡器权向量编码为 DNA 核苷酸单链，设置种群规模为 $M$ 个个体。以整个种群为搜索空间，对于种群

中的单链个体，生成与之配对的互补链，从而形成双链 DNA 个体。将经过正交小波变换后的信号作为 DNA 遗传算法的输入信号，分别计算每一个个体的两条互补的单链适应度值，选取适应度值最大的单链序列作为该个体的代表链。根据代表链个体适应度值大小将所有个体进行排序，前一半个体为优质种群，后一半个体为劣质种群，并且将种群中个体适应度值最大的个体作为精英个体保留。

步骤 2：构建禁忌表并设置当前种群的适应度平均值为渴望水平。然后按图 7.10 所示的禁忌交叉操作流程执行交叉操作并不断更新渴望水平。其中，置换交叉、转位交叉和重构交叉操作的顺序是，首先在优质种群中随机选择两个个体作为父体用于执行交叉操作；然后对被选中的父体分别以概率 $p_1$ 和 $p_2$ 执行置换交叉操作和转位交叉操作。如果置换交叉操作和转位交叉操作均未被执行，则按概率 $p_3$ 执行重构交叉操作，每次交叉操作产生的不放回原种群。重复以上交叉操作直到产生 $M/2$ 新个体，然后将这 $M/2$ 新个体放入到原种群中，得到 $3M/2$ 个个体。

步骤 3：在新得到的子代个体中对每个个体按 7.1.1 节执行变异操作。对于种群中每个个体，都执行一次变异操作，然后用变异后产生的个体取代原个体。变异操作完成后，重复执行 $M-1$ 次联赛选择操作，选择出 $M-1$ 个个体，与原来的精英个体一起组成新种群，最后计算每个个体的适应度值，将适应度值最大的个体作为最优个体。种群进化代数加 1。此时种群中的个体都是单链 DNA 序列，为每一个单链个体生成对应的互补链，形成双链 DNA 序列，并分别计算原个体单链和互补链的适应度值。

步骤 4：判断是否达到进化终止条件。如果当前进化代数小于最大进化代数，则继续选择代表链并且对新种群分组，完成步骤 2～步骤 4，直到满足进化终止条件为止。

步骤 5：如果满足进化条件，则输出适应度值最大的个体，并将该个体解码后的值作为最优值对衡器权向量进行初始化。

具体操作流程，如图 7.11 所示。

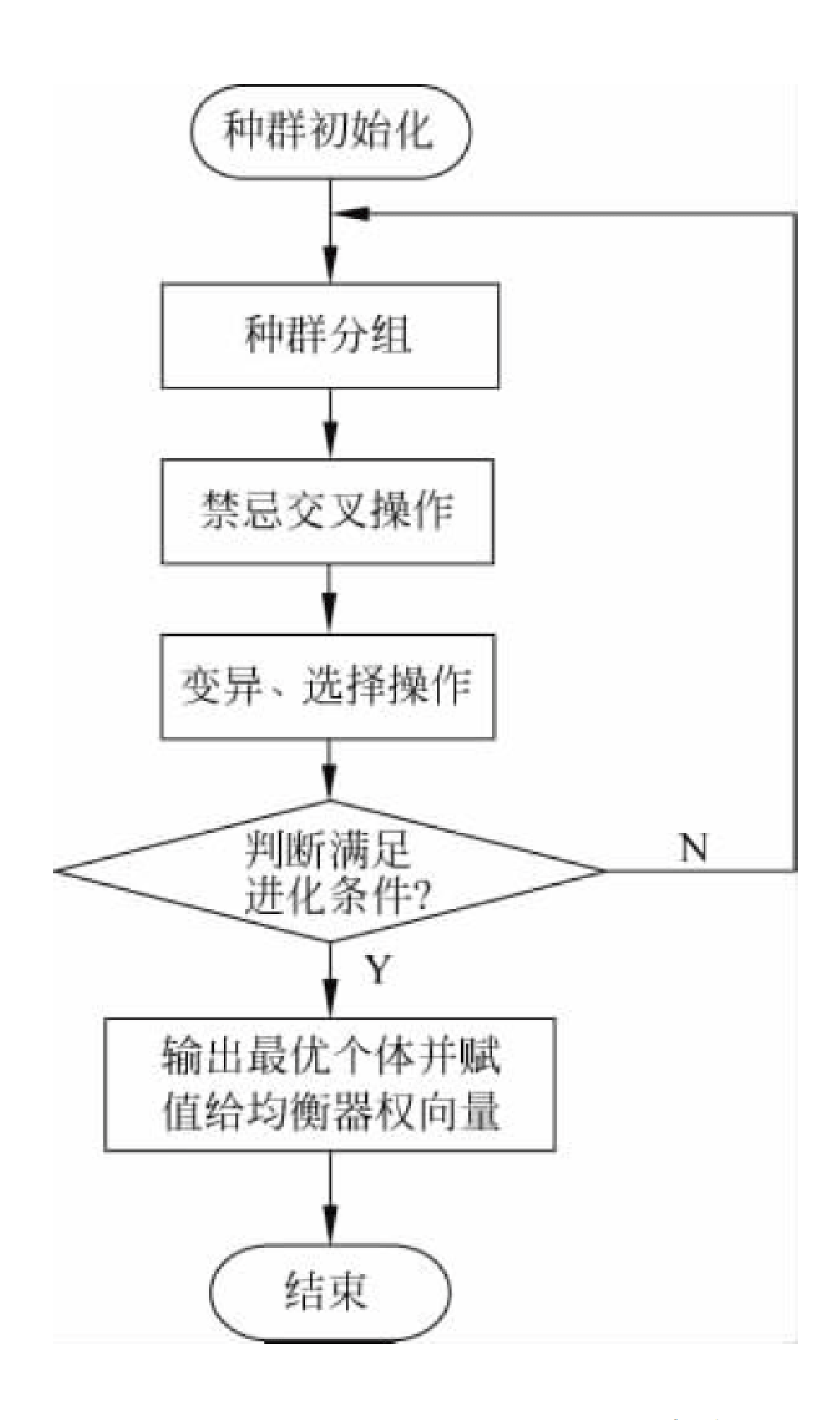



图 7.11 TSDNAGA-WTCMA 流程图

5. 算法仿真

为了验证 TSDNAGA-WTCMA 的有效性，以 WTCMA 和 CMA 为比较对象，进行仿真实验。

**【实验 7.2】** 当信道 $\boldsymbol{h}=[0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$时，信道噪声为高斯白噪声，发射信号为 8PSK，均衡器权长为 16，步长 $\mu_{\text{CMA}}=0.0022$，$\mu_{\text{WTCMA}}=0.0021$，$\mu_{\text{TSDNAGA-CMA}}=0.0029$，信噪比为 20dB，中心抽头初始化，训练样本个数 $N=8000$。对信道的输入信号采用 DB2 正交小波分解，功率初始值设置为 4，遗忘因子 $\beta=0.99$。TSDNAGA-WTCMA 种群规模取 50，交叉操作按图 7.10 所示禁忌搜索交叉操作执行，终止进化代数为 100。

当信道 $\boldsymbol{h}=[0.9656 \quad -0.0906 \quad 0.0578 \quad 0.2368]$时，信道噪声为高斯白噪声，发射信号为 16PSK，均衡器权长为 16，步长 $\mu_{\text{CMA}}=0.000\,13$，

$\mu_{\text{WTCMA}}=0.00028$，$\mu_{\text{TSDNAGA-WTCMA}}=0.00035$，信噪比为 20dB，训练样本个数 $N=8000$。对信道的输入信号采用 DB2 正交小波分解，功率初始值设置为 4，遗忘因子 $\beta=0.99$。TSDNAGA-WTCMA 种群规模取 50，交叉操作按图 7.10 所示禁忌搜索交叉操作执行，终止进化代数为 100。

在保证眼图完全清晰睁开的前提下，500 次蒙特卡罗仿真结果如图 7.12 所示。

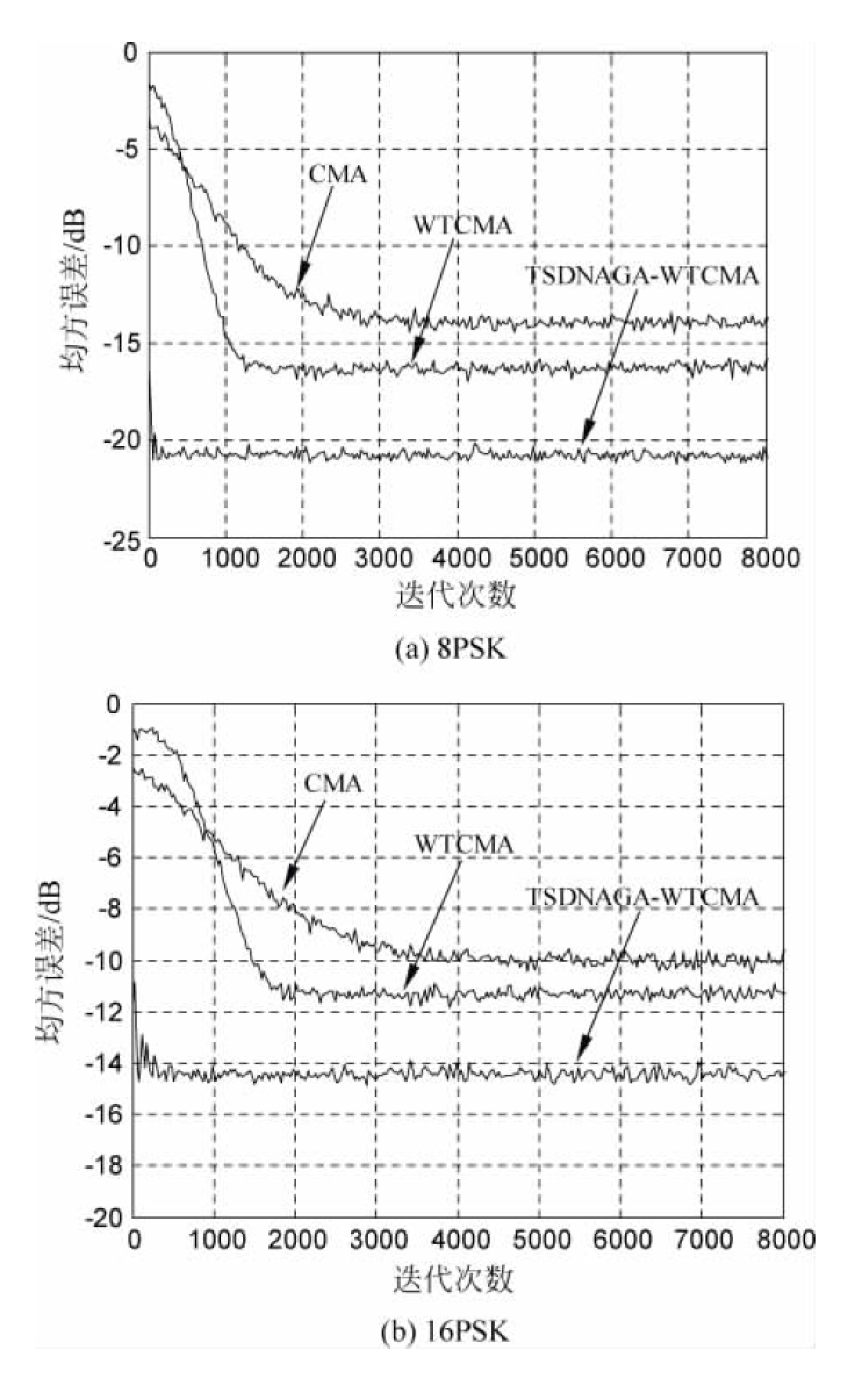



图 7.12 仿真结果

图 7.12(a)表明,DNAGA-WTCMA 收敛速度比 WTCMA 快约 1200 步;比 CMA 快约 2000 步;在稳态误差上,DNAGA-WTCMA 的稳态误差比 WTCMA 的稳态误差小约 4dB,比 CMA 的稳态误差小约 6dB。图 7.12(b)表明,TSDNAGA-WTCMA 的收敛速度比 WTCMA 快约 1800 步,比 CMA 快约 3500 步;在稳态误差上,TSDNAGA-WTCMA 的稳态误差比 WTCMA 的稳态误差小约 2.5dB,比 CMA 的稳态误差小约 3.5dB。

### 7.3.3 基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波多模盲均衡算法

多模盲均衡算法(MMA)利用均衡器输入信号的幅度和相位信息,有效克服了 CMA 单一判决圆造成的误差[25-26]。将正交小波变换应用于多模盲均衡算法中,降低了均衡器输入信号的相关性、改善了 MMA 收敛性能。然而,CMA 和 MMA 都是采用梯度搜索法对均衡器权向量进行更新的,容易陷入局部收敛,难以获得全局最优解。为了提高盲均衡算法对高阶多模信号的均衡性能,本节将 DNA 遗传算法与 WTMMA 相结合,提出了一种基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波多模盲均衡算法(TS algorithm and adaptive double chain DNA Genetic Algorithm based WTMMA,TSDNAGA-WTMMA)。

#### 1. 算法原理

为了提高盲均衡算法的收敛性能,将正交小波变换应用于 MMA 中,降低了输入信号的自相关性,改善了算法的均衡性能。将 DNA 遗传算法与正交小波多模盲均衡算法相结合,就得到 TSDNAGA-WTMMA,其原理如图 7.13 所示。

均衡器权向量为

$$\boldsymbol{f}(k) = \boldsymbol{f}_{\mathrm{R}}(k) + \mathrm{j}\boldsymbol{f}_{\mathrm{I}}(k) \tag{7.3.6}$$

经过正交小波变换后均衡器的输入信号为

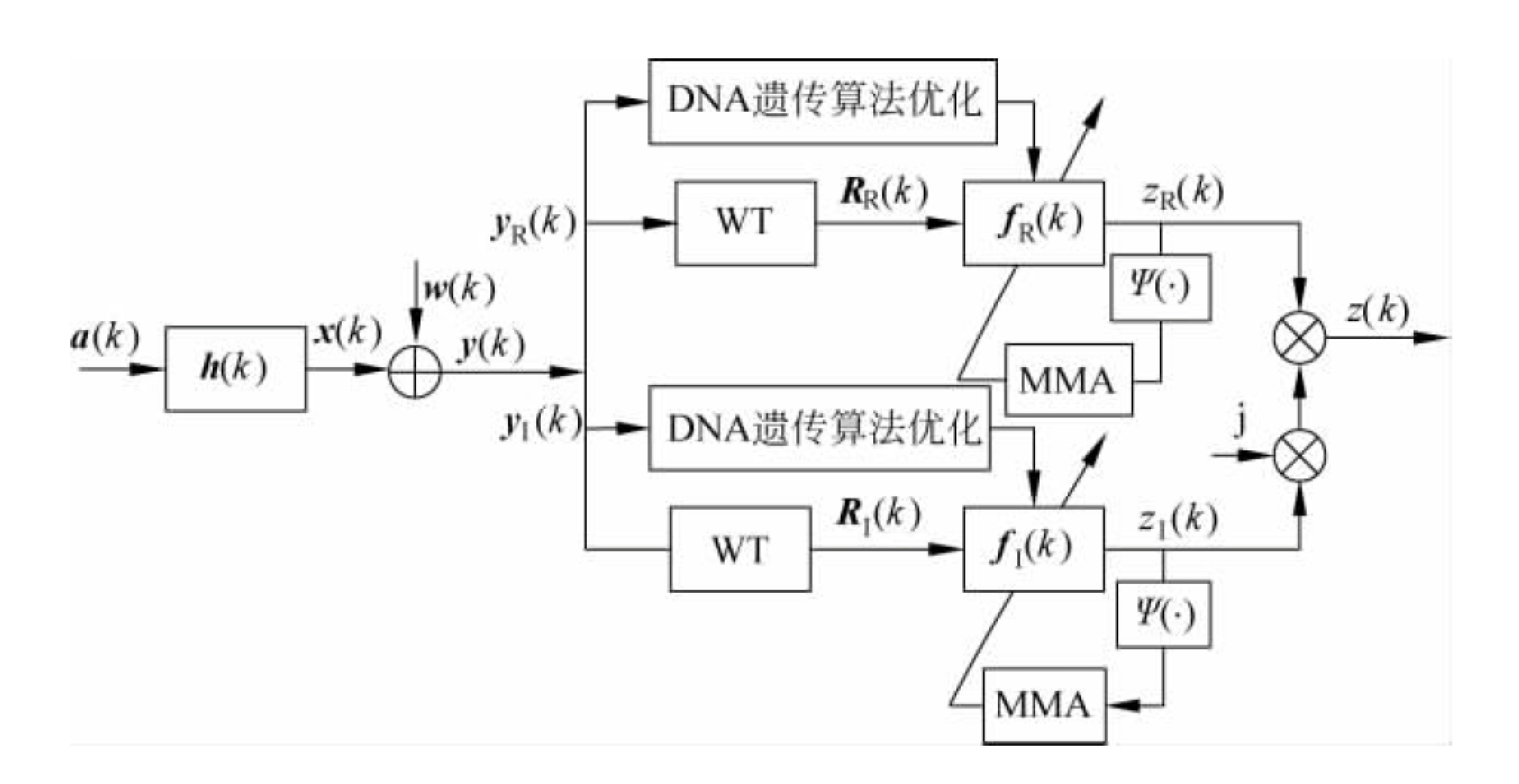



图 7.13　TSDNAGA-WTMMA 原理

$$\boldsymbol{R}(k)=\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}\boldsymbol{R}_{\mathrm{I}}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{V}\boldsymbol{y}_{\mathrm{I}}(k)) \tag{7.3.7}$$

式中，$\boldsymbol{V}$ 为正交小波变换矩阵，$\boldsymbol{R}_{\mathrm{R}}(k)$和 $\boldsymbol{R}_{\mathrm{I}}(k)$分别为输入信号 $\boldsymbol{R}(k)$的实部与虚部，即

$$\boldsymbol{R}_{\mathrm{R}}(k)=[u_{\mathrm{R}1,0}(k),u_{\mathrm{R}1,1}(k),\cdots,u_{\mathrm{R}J,k_J}(k),s_{\mathrm{R}J,0}(k),\cdots,s_{\mathrm{R}J,k_J}(k)]^{\mathrm{T}} \tag{7.3.8a}$$

$$\begin{cases} u_{\mathrm{R}j,m}(k)=\sum\limits_{l=0}^{L-1}y_{\mathrm{R}}(k-l)\varphi_{j,m}(l) \\ s_{\mathrm{R}J,m}(k)=\sum\limits_{l=0}^{L-1}y_{\mathrm{R}}(k-l)\varphi_{J,m}(l) \end{cases} \tag{7.3.8b}$$

$$\boldsymbol{R}_{\mathrm{I}}(k)=[u_{\mathrm{I}1,0}(k),u_{\mathrm{I}1,1}(k),\cdots,u_{\mathrm{I}J,k_J}(k),s_{\mathrm{I}J,0}(k),\cdots,s_{\mathrm{I}J,k_J}(k)]^{\mathrm{T}} \tag{7.3.9a}$$

$$\begin{cases} u_{\mathrm{I}j,m}(k)=\sum\limits_{l=0}^{L-1}y_{\mathrm{I}}(k-l)\varphi_{j,m}(l) \\ s_{\mathrm{I}J,m}(k)=\sum\limits_{l=0}^{L-1}y_{\mathrm{I}}(k-l)\varphi_{J,m}(l) \end{cases} \tag{7.3.9b}$$

式中，$u_{j,m}(k)$，$s_{J,m}(k)$分别为相应的小波和尺度变换系数。

均衡器的输出为

$$\begin{aligned} z(k)&=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k) \\ &=\boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{R}}(k)+\mathrm{j}(\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{I}}(k)) \end{aligned} \tag{7.3.10}$$

式中，$\boldsymbol{f}_{\mathrm{R}}(k)$，$\boldsymbol{f}_{\mathrm{I}}(k)$分别表示均衡器权向量的实部与虚部。

WTMMA 的代价函数为

$$J_{\mathrm{WTMMA}}=E\{(z_{\mathrm{R}}^{2}(k)-R_{\mathrm{R,WTMMA}}^{2})^{2}+(z_{\mathrm{I}}^{2}(k)-R_{\mathrm{I,WTMMA}}^{2})^{2}\} \quad (7.3.11)$$

式中，$R_{\mathrm{R,WTMMA}}^{2}=\dfrac{E[a_{\mathrm{R}}^{4}(k)]}{E[a_{\mathrm{R}}^{2}(k)]}$，表示 $a(k)$的实部模值；$R_{\mathrm{I,WTMMA}}^{2}=\dfrac{E[a_{\mathrm{I}}^{4}(k)]}{E[a_{\mathrm{I}}^{2}(k)]}$，表示 $a(k)$的虚部模值。

均衡器的误差函数为

$$\begin{cases} e_{\mathrm{R,WTMMA}}(k)=z_{\mathrm{R}}(n)(z_{\mathrm{R}}^{2}(k)-R_{\mathrm{R,WTMMA}}^{2}) \\ e_{\mathrm{I,WTMMA}}(k)=z_{\mathrm{I}}(n)(z_{\mathrm{I}}^{2}(k)-R_{\mathrm{I,WTMMA}}^{2}) \end{cases} \quad (7.3.12)$$

均衡器权向量的更新公式为

$$\begin{cases} \boldsymbol{f}_{\mathrm{R}}(k+1)=\boldsymbol{f}_{\mathrm{R}}(k)-\mu\hat{\boldsymbol{R}}^{-1}(k)e_{\mathrm{R,WTMMA}}(k)\boldsymbol{R}_{\mathrm{R}}^{*}(k) \\ \boldsymbol{f}_{\mathrm{I}}(k+1)=\boldsymbol{f}_{\mathrm{I}}(k)-\mu\hat{\boldsymbol{R}}^{-1}(k)e_{\mathrm{I,WTMMA}}(k)\boldsymbol{R}_{\mathrm{I}}^{*}(k) \end{cases} \quad (7.3.13)$$

式中，$\hat{\boldsymbol{R}}^{-1}(k)=\mathrm{diag}[\sigma_{j,0}^{2}(k),\sigma_{j,1}^{2}(k),\cdots,\sigma_{J,k_j-1}^{2}(k),\sigma_{J+1,0}^{2}(k),\cdots,\sigma_{J+1,k_j-1}^{2}(k)]$，$\sigma_{j,k}^{2}(k)$、$\sigma_{J+1,k_j}^{2}(k)$分别表示对小波变换系数 $u_{j,m}(k)$和尺度变换系数 $s_{J,m}(k)$的平均功率估计，其递推公式为

$$\begin{cases} \sigma_{j,m}^{2}(k+1)=\beta\sigma_{j,m}^{2}(k)+(1-\beta)\mid u_{j,m}(k)\mid^{2} \\ \sigma_{J+1,m}^{2}(k+1)=\beta\sigma_{J+1,m}^{2}(k)+(1-\beta)\mid s_{J,m}(k)\mid^{2} \end{cases} \quad (7.3.14)$$

式中，$\beta$ 为平滑因子，且 $0<\beta<1$。为解决 WTMMA 容易陷入局部收敛的问题，采用禁忌搜索自适应双链 DNA 遗传算法对其初始权向量进行优化，就得到基于禁忌搜索自适应双链 DNA 遗传算法优化的正交小波多模盲均衡算法(TSDNAGA-WTMMA)。TSDNAGA-WTMMA 算法操作步骤如下。

步骤 1：首先按照 7.1.1 节的编码方式将均衡器权向量编码为 DNA 单链分子，设置种群规模为 $M$ 个个体。对于种群中的单链个体，生成与之配对的互补链，从而形成双链 DNA 个体。将经过小波变换后的信号作为 DNA 遗传算法的输入信号，分别计算每一个个体的两条互补的单链适应度值，选取适应度值最大的单链序列作为该个体的代表链。根据代表链个体适应度值大小将所有个体进行排序，前一半个体为优质种群，后一半个体为

劣质种群,并且将种群中个体适应度值最大的个体作为精英个体保留。

步骤 2:确定适应度函数。将均衡器权向量用 DNA 遗传算法编码表示成 DNA 核苷酸序列,设初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$,$M$ 为种群个体数量,$\boldsymbol{f}_m$,$1\leqslant m\leqslant M$ 对应一组均衡器权向量。

将 WTMMA 的代价函数作为 DNA 遗传算法的适应度函数。由于 DNA 遗传算法最终获得种群中适应度值最大的个体,而现在需要获得代价函数的极小值。因此,将适应度函数定义为代价函数的倒数,即

$$\mathrm{Fit}(\boldsymbol{f}_m)=\frac{b}{J_{\mathrm{WTMMA}}(\boldsymbol{f}_m)} \tag{7.3.15}$$

式中,$b$ 表示比例系数。

步骤 3:构建禁忌表并设置当前种群的适应度平均值为渴望水平。以整个种群作为搜索空间,按照图 7.10 所示的禁忌交叉操作流程执行交叉操作并不断更新渴望水平。其中,置换交叉、转位交叉和重构交叉操作的顺序是,首先在优质种群中随机选择两个个体作为父体用于执行交叉操作。然后对被选中的父体分别以概率 $p_1$ 和 $p_2$ 执行置换交叉操作和转位交叉操作;如果置换交叉操作和转位交叉操作均未被执行,则按概率 $p_3$ 执行重构交叉操作,每次交叉操作产生的个体不放回原种群。重复以上交叉操作直到产生 $M/2$ 新个体,然后将这 $M/2$ 新个体放入到原种群中,得到 $3M/2$ 个个体。

步骤 4:在新得到的子代个体中对每一个个体按 7.1.1 节执行变异操作。对于种群中的每一个个体,都执行一次变异操作,然后将变异后产生的个体取代原个体。变异操作完成后,重复执行 $M-1$ 次联赛选择操作,选出 $M-1$ 个个体,与原来的精英个体一起组成新种群;最后计算每个个体的适应度值,将适应度值最大的个体作为最优个体,种群进化代数加 1。此时种群中的个体都是单链 DNA 序列,为每一个单链个体生成对应的互补链,形成双链 DNA 序列,并分别计算原个体单链和互补链的适应度值。

步骤 5:判断是否达到进化终止条件。如果当前进化代数小于最大进化代数,则继续选择代表链并对新种群分组,完成步骤 2～步骤 4,直到满足进化终止条件为止。

步骤 6：如果满足进化条件，则输出适应度值最大的个体，并将该个体解码后的值作为最优值对衡器权向量进行初始化。

TSDNAGA 优化 WTMMA 流程如图 7.14 所示。

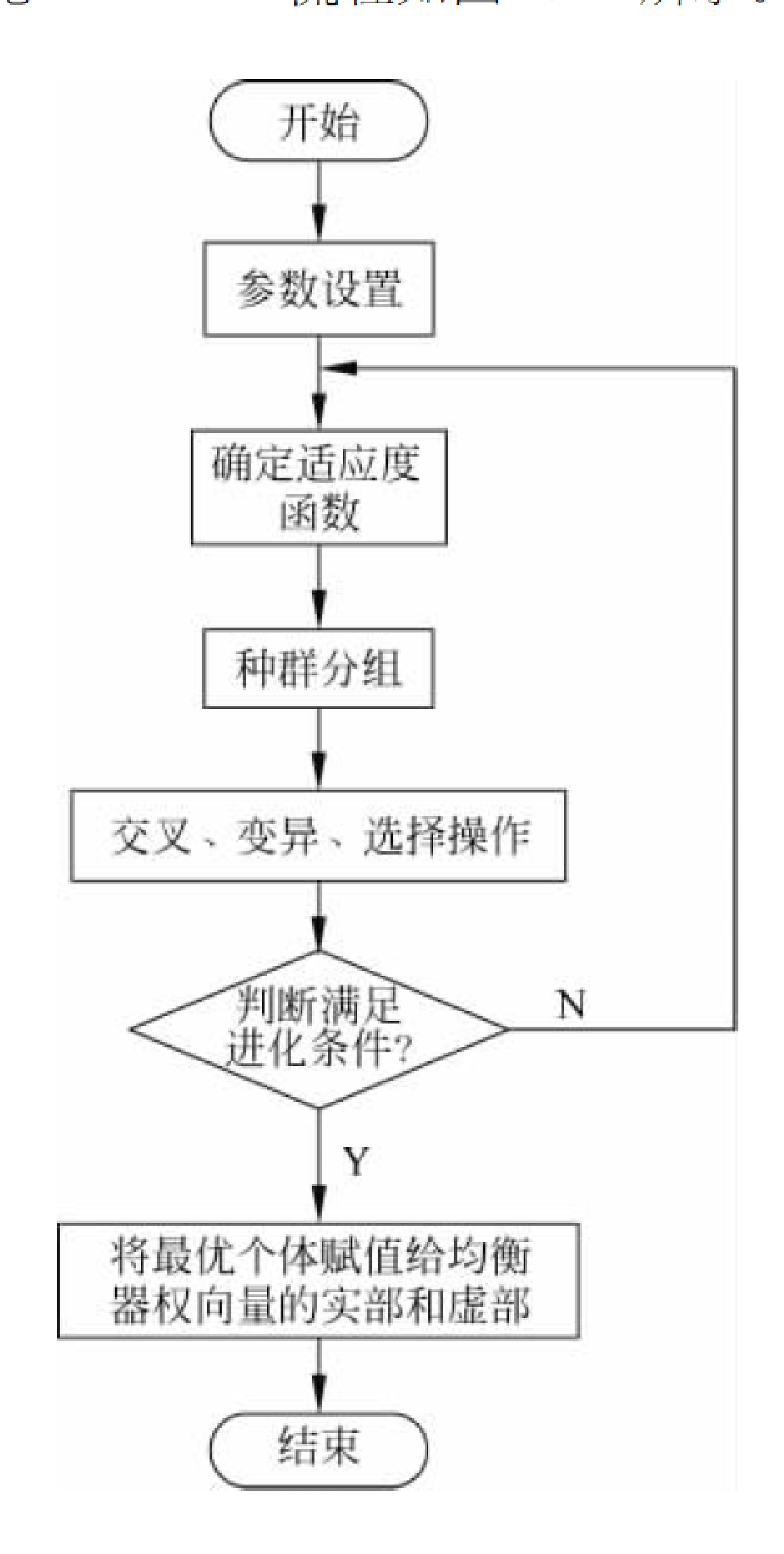



图 7.14 TSDNAGA 优化 WTMMA 流程

2. 算法仿真

**【实验 7.3】** 混合相位水声信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$，发射信号为 64QAM，信噪比为 25dB，均衡器权长为 16，第 4 个抽头系数设置为 1，其余为 0；步长 $\mu_{\text{MMA}}=0.000\,04$，$\mu_{\text{WTMMA}}=0.000\,03$，$\mu_{\text{TSDNAGA-WTMMA}}=0.000\,033$；对信道的输入信号采用 DB2 正交小波分解，功率初始值设置为 4，遗忘因子 $\beta=0.99$。DNAGA-WTCMA 种群规模取 50，终止进化代数为 100。

采用信道 $\boldsymbol{h}=[0.9656 \quad -0.0906 \quad 0.0578 \quad 0.2368]$时，发射信号为128QAM信号，信噪比为25dB，均衡器权长为16，第4个抽头系数设置为1，其余为0；步长 $\mu_{\mathrm{MMA}}=0.65\times10^{-6}$，$\mu_{\mathrm{WTMMA}}=0.54\times10^{-6}$，$\mu_{\mathrm{TSDNAGA\text{-}WTMMA}}=0.4\times10^{-6}$；对信道的输入信号采用DB2正交小波分解，功率初始值设置为4，遗忘因子 $\beta=0.99$。DNAGA-WTCMA种群规模取50，终止进化代数为100。

在保证眼图完全清晰睁开的前提下，500次蒙特卡罗仿真结果如图7.15所示。

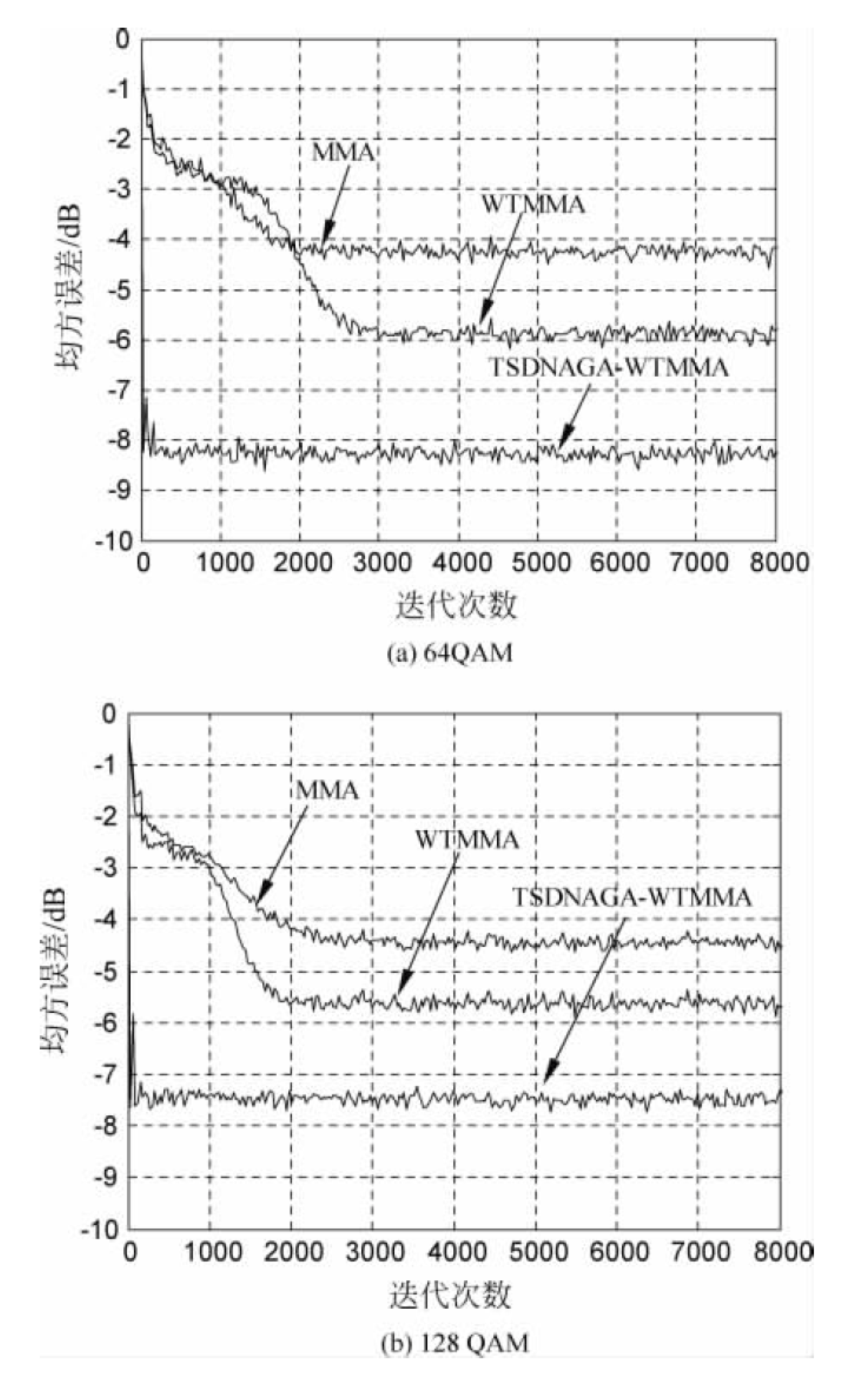



图7.15　均方误差迭代曲线

图 7.15(a) 表明，TSDNAGA-WTMMA 的收敛速度比 MMA 和 WTMMA 更快；在稳态误差上，TSDNAGA-WTMMA 的稳态误差比 WTMMA 的稳态误差小约 2dB，比 MMA 的稳态误差小约 4dB。图 7.15(b) 表明，TSDNAGA-WTMMA 的收敛速度最快，比 WTMMA 快约 1700 步，比 MMA 快约 2000 步；在稳态误差上，TSDNAGA-WTMMA 的稳态误差比 WTMMA 的稳态误差小约 1.5dB，比 MMA 的稳态误差小约 3dB。

## 7.4 基于多种群进化禁忌搜索 DNA 遗传算法优化的正交小波分数间隔盲均衡算法

DNA 遗传算法在搜索最优解时具有较强的随机性，它以交叉操作和变异操作为指导来确定进化方向。维持种群的多样性是算法能够搜索到全局最优解的关键因素，全局搜索能力和局部搜索能力是衡量搜索算法优劣的标准之一。DNA 遗传算法在进化过程中，由于交叉、变异和选择操作的随机性，有可能出现种群多样性下降、早熟收敛并陷入局部最优，无法获得全局最优解。在一般的 DNA 遗传算法中，所有的操作都是在一个种群中执行的，在搜索最优解过程中，不仅要求算法有较强的全局搜索能力，而且还需要有准确的局部搜索能力以获得精确的全局最优值，只有同时兼顾全局搜索和局部搜索才能提高搜索效率。然而，增强全局搜索和局部搜索能力是一对矛盾，即增强全局搜索能力会在一定程度上削弱局部搜索能力，同样，提高局部搜索能力也会影响全局搜索能力。针对这种情况，很多学者提出了多种群进化算法[27-28]。

为了提高 DNA 遗传算法的搜索性能，本节采用多种群 DNA 遗传算法，将种群分为多个种群以实现全局搜索和局部搜索的分离，并将该算法与分数间隔盲均衡算法相结合，研究了一种基于多种群进化 DNA 遗传算法优化的正交小波分数间隔多模盲均衡算法[14]。

### 7.4.1 多种群禁忌搜索DNA遗传算法

#### 1. 多种群DNA遗传算法

多种群DNA遗传算法的思想是：首先按照种群个体适应度值将种群分为主种群和辅助种群，其中主种群包含适应度值较大的个体，辅助种群包含适应度值较小的个体。每个子种群分别按照不同的进化目的独立进化，主种群主要侧重于局部搜索，辅助种群侧重于全局搜索；在各个子种群同时进化的过程中，需要完成个体迁移操作，把每个辅助种群的最优个体作为迁移个体迁移到主种群中，从而保证种群间的信息传递[29-31]。

首先设置种群规模，为了便于将种群分组，种群的规模取6的整数倍。将种群按适应度值从大到小排列，适应度值最大的1/3部分种群记为主种群，适应度值在中间的1/3部分种群记为辅助种群1，适应度值最小的1/3部分种群记为辅助种群2。对于不同的种群，分别采用不同的进化操作。对于主种群来说，在进化过程中需要注重局部搜索，在变异操作过程中，个体高位部分的变异概率已知处于较低的水平，而低位部分的变异概率则处于较高的水平；对于辅助种群1，由该种群的适应度平均值介于主种群和辅助种群2之间，因此该种群在进化过程中既注重全局搜索，又要注重局部搜索，因此采用自适应变异概率操作；对于辅助种群2，由于该种群所包含的个体是整个种群中适应度值最小的个体，因此该种群在进化过程中主要侧重于全局搜索。在变异操作过程中，该种群个体的高位部分采用较高的变异概率，而低位部分采用较低的变异概率，以加强全局搜索能力。对于这三个独立的并行种群，在进化过程中需要将辅助种群的最优个体不断地迁移到主种群中，以保证主种群搜索的正确性并且实现种群间信息交换。在每个种群的进化过程中，采用相应的禁忌交叉操作和选择操作，以进一步提高算法的性能。在种群进化到一定的代数后，将子种群合并成一个种群，继续按照上述步骤完成进化操作。虽然主种群与辅助种群是相互独立进化的，但在操作过程中，主种群与辅助种群必须要相互协调，以保证每个种群搜索

的正确性。

多种群禁忌搜索 DNA 遗传算法步骤如下。

步骤 1：将初始种群按适应度值大小分为主种群、辅助种群 1 和辅助种群 2。

步骤 2：在不同的种群中分别采用相应的变异操作、禁忌交叉操作、选择操作使每个子种群独立的进化。

步骤 3：每个种群进化完一代，则对子种群的最优个体进行迁移操作。

步骤 4：判断是否达到所设定的代数，若满足则对三个子种群进行合并，然后返回步骤 1；否则返回步骤 2。

多种群进化禁忌搜索 DNA 遗传算法流程，如图 7.16 所示。在该算法中，不同的子种群分别使用同一个代价函数确定种群的适应度函数，每个种群采用不同的操作参数，以实现不同的搜索目的。

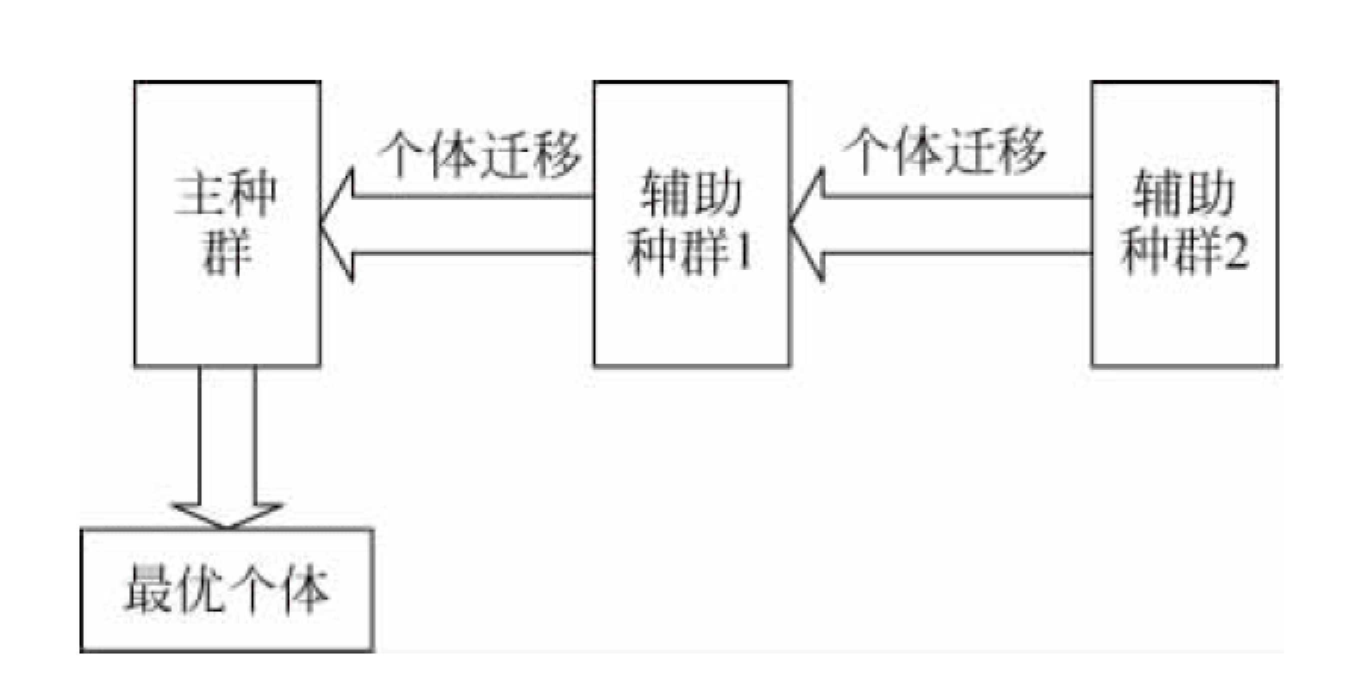



图 7.16 多种群进化的禁忌搜索 DNA 遗传算法流程

2. 多种群变异操作

变异操作是维持种群多样性的关键因素。本节中，多种群 DNA 遗传算法对不同的种群采用的变异操作有所不同。对于辅助种群 1，为了使其同时具有全局搜索和局部搜索能力，其变异概率规则如式(7.1.7)和式(7.1.8)所示，变异曲线如图 7.6 所示。对于主种群，需要种群具有较强的局部搜索能力，以准确寻找最优解，所以需要个体高位部分具有较低的变异概率，而低位部分具有较高的变异概率，其变异概率如式(7.4.1)和式(7.4.2)所示。变异概率曲线，如图 7.17 所示。

$$p_{ml}=\begin{cases}a_1+\dfrac{b_1}{1+\exp[a(g-g_0)]}, & g<\dfrac{G}{2}\\ a_1+\dfrac{b_1}{1+\exp[-a(g-g_0)]}, & g>\dfrac{G}{2}\end{cases} \tag{7.4.1}$$

$$p_{mh}=\begin{cases}a_1+\dfrac{b_1}{1+\exp[-a(g-g_0)]}, & g<\dfrac{G}{2}\\ a_1+\dfrac{b_1}{1+\exp[a(g-g_0)]}, & g>\dfrac{G}{2}\end{cases} \tag{7.4.2}$$

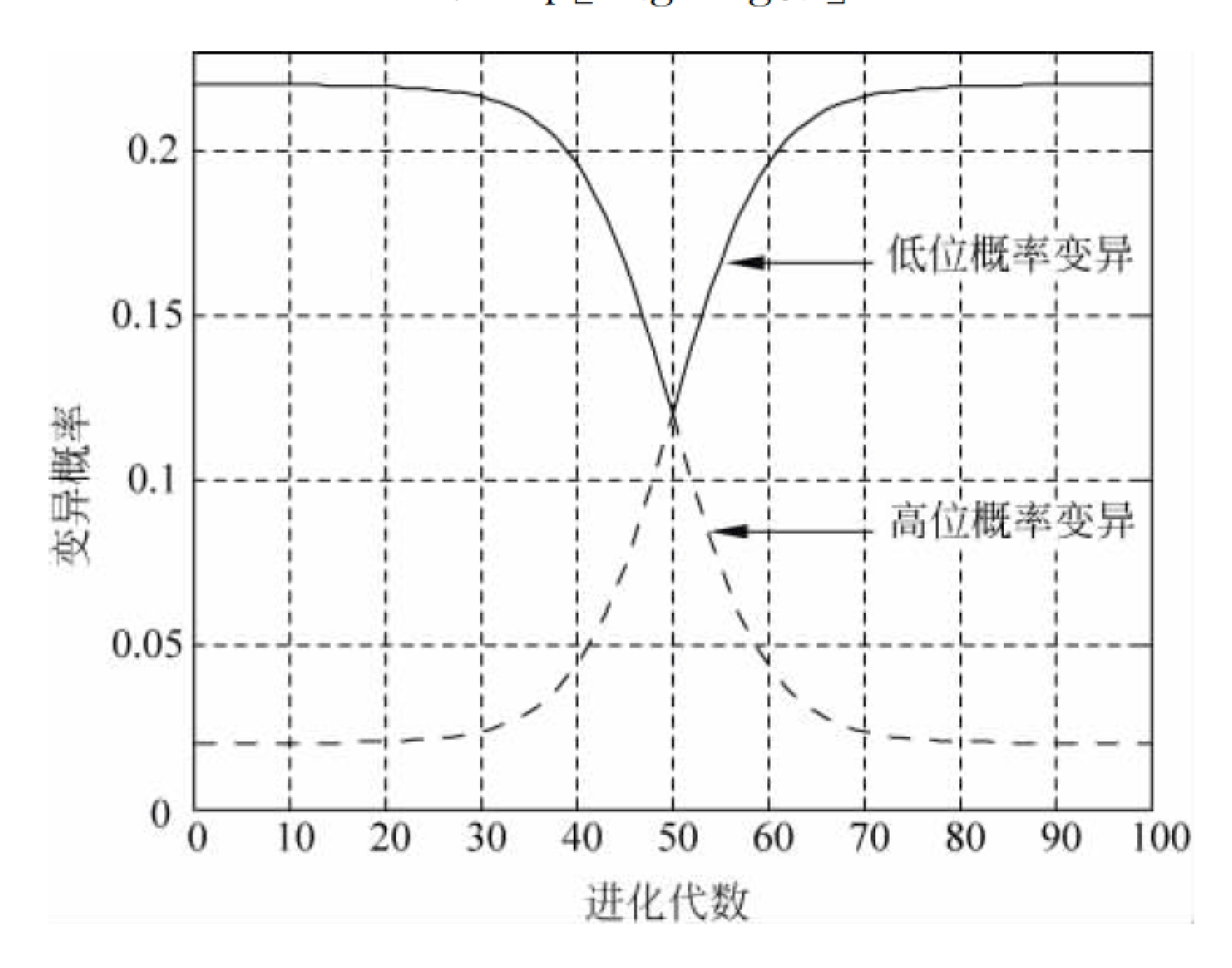



图 7.17 主种群变异概率曲线

对于辅助种群 2,由于该种群所包含的个体是整个种群中适应度值最小的个体,因此该种群在进化过程中侧重于全局搜索。在变异操作过程中,该种群个体的高位部分采用较高的变异概率,而低位部分采用较低的变异概率,以加强全局搜索能力。其变异概率如式(7.4.3)和式(7.4.4)所示。变异概率曲线,如图 7.18 所示。

$$p_{ml}=\begin{cases}a_1+\dfrac{b_1}{1+\exp[-a(g-g_0)]}, & g<\dfrac{G}{2}\\ a_1+\dfrac{b_1}{1+\exp[a(g-g_0)]}, & g>\dfrac{G}{2}\end{cases} \tag{7.4.3}$$

$$p_{mh}=\begin{cases}a_1+\dfrac{b_1}{1+\exp[a(g-g_0)]}, & g<\dfrac{G}{2}\\ a_1+\dfrac{b_1}{1+\exp[-a(g-g_0)]}, & g>\dfrac{G}{2}\end{cases} \tag{7.4.4}$$

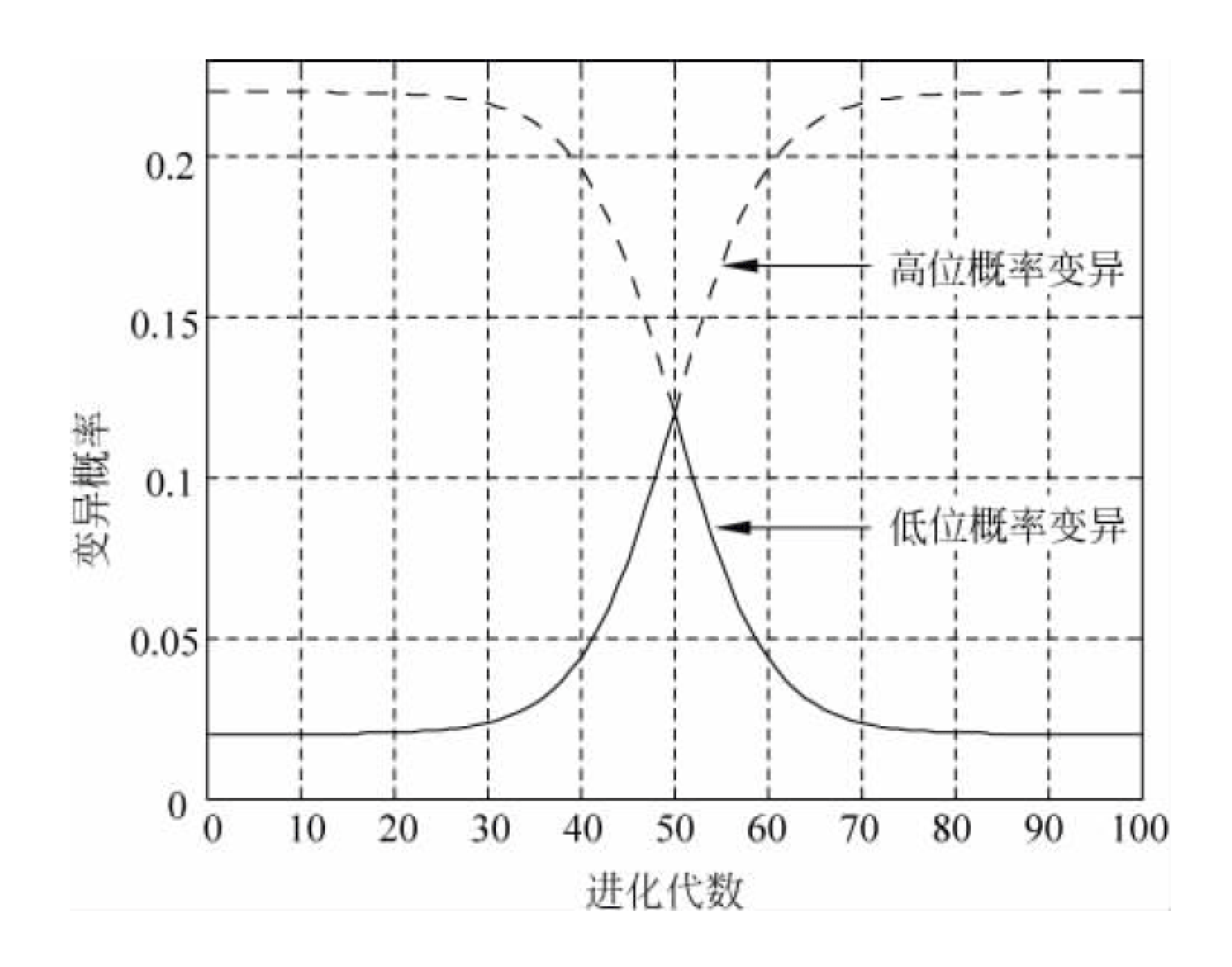



图 7.18　辅助种群 2 变异概率曲线

3. 多种群选择操作

本节使用联赛选择方法产生新种群。联赛选择就是在种群中随机选择一定量的个体，挑选出适应度值最大的个体进入下一代，再将剩余的个体放回原种群，重复以上过程，直到子代个体数量达到要求值。另外，为了防止将种群中适应度值最大的个体丢失掉，采用精英保留机制，即直接把父代种群中适应度值最大的个体复制到下一代。

4. 多种群禁忌搜索 DNA 遗传算法操作

根据 DNA 编码规则及交叉、变异和选择操作，多种群禁忌搜索 DNA 遗传算法的操作步骤归纳如下。

步骤 1：设置相关参数。设置最大进化代数 $G_{\max}$，种群规模 $M$，DNA 序列编码长度 $L$，算法终止阈值 $\Delta$。

步骤 2：确定适应度函数并计算适应度值。初始化种群，随机生成 $M$ 个长度为 $N\times L$ 的 DNA 序列构成初始种群。根据 7.1.1 节的解码规则将每个 DNA 序列解码，然后代入到适应度函数中，计算每个 DNA 个体的适应度值。

步骤 3：将 DNA 种群分成主种群、辅助种群 1 和辅助种群 2。以整个种群作为搜索空间，根据个体适应度值大小将所有个体进行排序，将种群个体平均分成 3 份，分别对应主种群、辅助种群 1 和辅助种群 2，称为子种群。在每个子种群中，将种群个体分为优质种群和劣质种群，并将每个子种群中个体适应度值最大的个体作为精英个体保留。

步骤 4：对种群中的个体执行禁忌交叉操作。对于每个子种群，分别按 7.3.1 节的内容确定相应种群的交叉操作概率，然后按照图 7.10 的禁忌交叉操作流程执行交叉操作并更新渴望水平。其中，置换交叉、转位交叉和重构交叉操作的顺序是，首先在优质种群中随机选择两个个体作为父体用于执行交叉操作，然后对被选中的父体分别以概率 $p_1$ 和 $p_2$ 执行置换交叉操作和转位交叉操作；如果置换交叉操作和转位交叉操作均未被执行，则按概率 $p_3$ 执行重构交叉操作，每次交叉操作产生的个体不放回原种群。重复以上交叉操作直到产生 $M/6$ 新个体，然后将这 $M/6$ 新个体放入到原子种群中，得到具有 $M/2$ 个个体的子种群。

步骤 5：对主种群和辅助种群分别执行变异操作。对每个子种群得到的具有 $M/2$ 个个体的混合种群执行变异操作，主种群个体变异概率采用式(7.4.1)和式(7.4.2)表示的变异操作概率，辅助种群 1 变异概率采用 7.1.1 节所述自适应动态变异操作，辅助种群 2 变异概率采用式(7.4.3)和式(7.4.4)表示的变异概率，用变异后的个体取代原个体。变异操作完成后，对每个子种群重复执行 $M/3-1$ 次联赛选择操作，挑选出 $M/3-1$ 个个体，与精英个体一起组成种群规模为 $M/3$ 的新种群，计算每个子种群中个体的适应度值，选择适应度值最大的个体作为子种群的最优个体。对于这三个并行遗传操作的子种群，分别进行优质个体迁移操作，用辅助种群 2 中的最优个体替换辅助种群 1 中的最差个体，用辅助种群 1 中的最优个体替换主种群中的最差个体，种群优质个体迁移操作完成后，种群进化代数加 1。

步骤 6：判断是否满足子种群合并条件。若不满足条件，则子种群继续进行独立的禁忌交叉、变异和选择操作；若满足种群合并条件，则将各种子

群合并成一个种群。

步骤 7：判断进化条件。如果达到设置的最大进化代数 $G_{max}$ 或者当前最优解的个体适应度值变化小于阈值 $\Delta$，则将主种群中适应度值最大的个体作为最优个体输出；否则，返回步骤 2。多种群禁忌搜索 DNA 遗传算法流程，如图 7.19 所示。

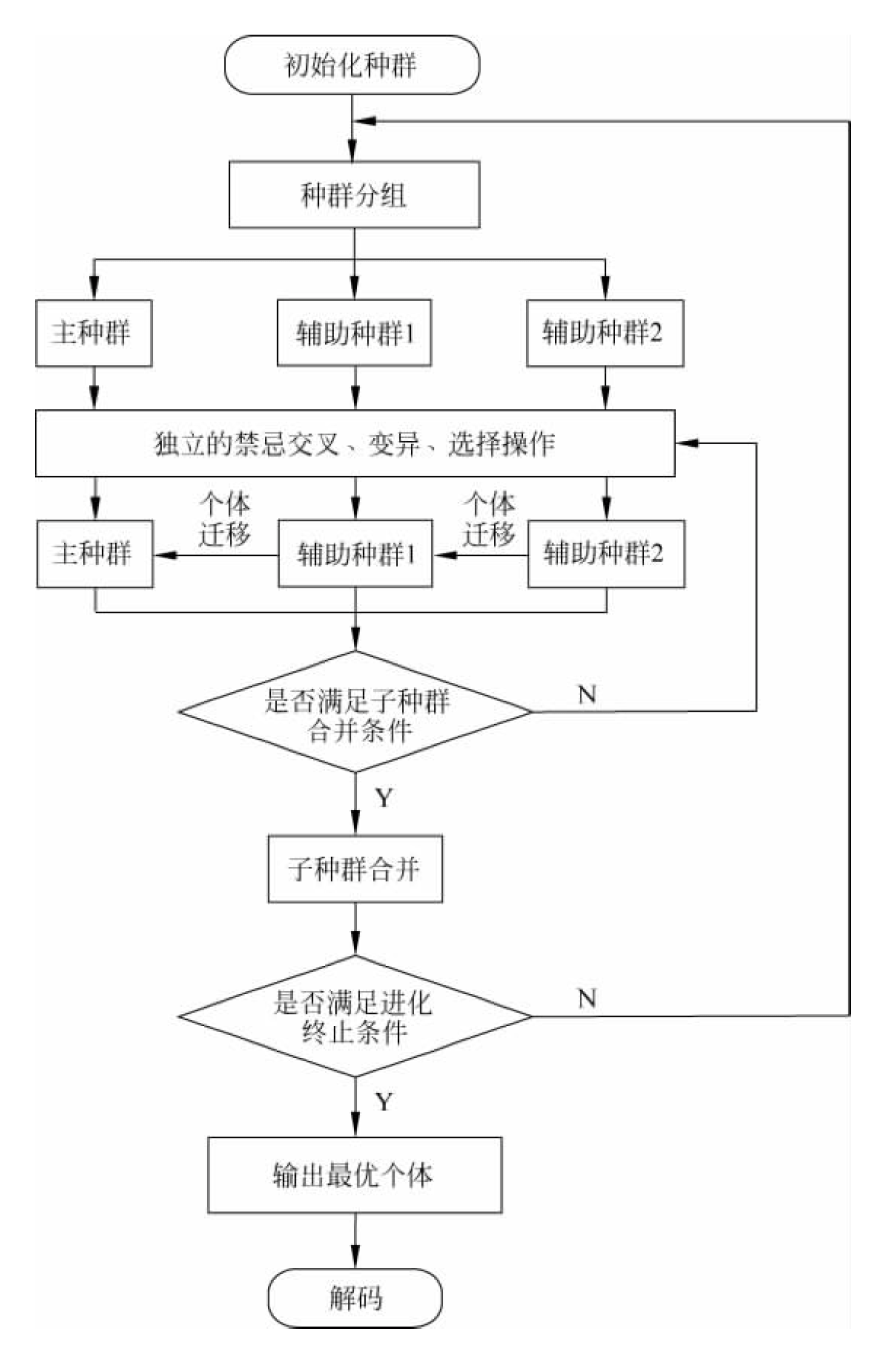



图 7.19　多种群禁忌搜索 DNA 遗传算法流程

### 7.4.2　多种群禁忌搜索 DNA 遗传算法优化分数间隔正交小波多模盲均衡算法

将多种群进化禁忌搜索 DNA 遗传算法引入到 T/P 分数间隔正交小波多模盲均衡算法中，得到基于多种群禁忌搜索 DNA 遗传算法优化的分数间隔正交小波多模盲均衡算法（Multiple population evolutionary Tabu Search algorithm and DNA genetic algorithm based WTMMA，MTSDNAGA-T/P-FSE-WTMMA）[14]，其原理如图 7.20 所示。

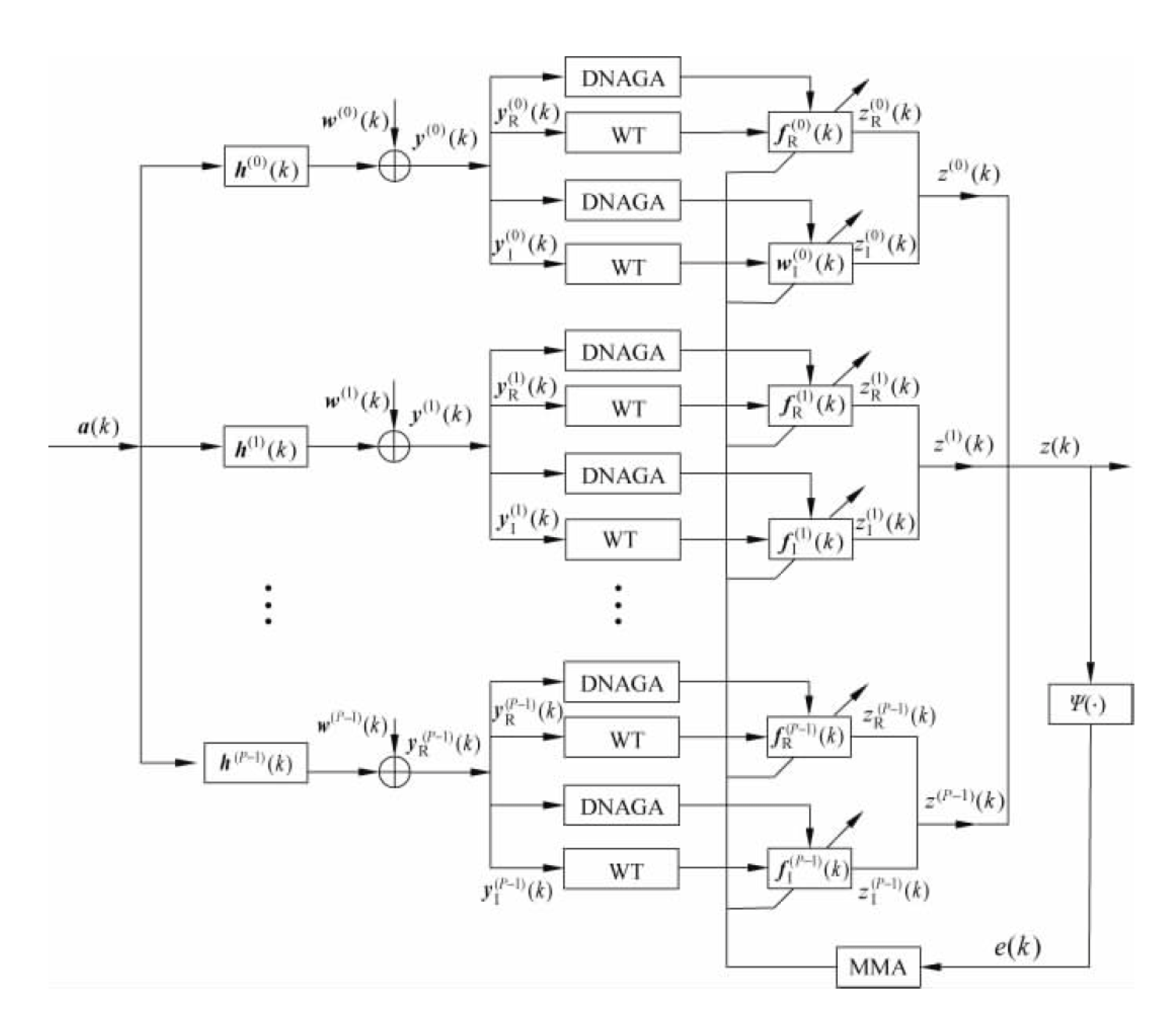



图 7.20　MTSDNAGA-T/P-FSE-WTMMA 原理

图 7.20 中，$\boldsymbol{a}(k)$是发射信号；$\boldsymbol{h}^{(p)}(k)$为第 $p$ 路子信道的脉冲响应；$\boldsymbol{w}^{(p)}(k)$为第 $p$ 路子信道的高斯白噪声；$\boldsymbol{y}^{(p)}(k)$为第 $p$ 路均衡器的输入信

号；$\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)$和 $\boldsymbol{f}_{\mathrm{I}}^{(p)}(k)$为第 $p$ 路子均衡器权向量的实部和虚部，每一路子均衡器权长为 $L_f$，且 $L_f=L/4$；$z^{(p)}(k)$为第 $p$ 路子信道均衡器的输出信号；$\Psi(\cdot)$是误差生成函数；$z(k)$为均衡器总输出；$e(k)$为误差信号。

设发射信号 $\boldsymbol{a}(k)$是以 $T$ 为周期的序列，第 $p$ 路子信道均衡器的输入信号为

$$\boldsymbol{y}^{(p)}(k)=\boldsymbol{h}^{(p)}(k)\boldsymbol{a}(k)+\boldsymbol{w}^{(p)}(k)=\boldsymbol{y}_{\mathrm{R}}^{(p)}(k)+\mathrm{j}\boldsymbol{y}_{\mathrm{I}}^{(p)}(k) \tag{7.4.5}$$

第 $p$ 路子均衡器权向量为

$$\boldsymbol{f}^{(p)}(k)=\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)+\mathrm{j}\boldsymbol{f}_{\mathrm{I}}^{(p)}(k) \tag{7.4.6}$$

式中，$\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)$和 $\boldsymbol{f}_{\mathrm{I}}^{(p)}(k)$为第 $p$ 路子均衡器权向量的实部和虚部

$$\begin{cases}\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)=[f_{\mathrm{R}}^{(p)}(0),f_{\mathrm{R}}^{(p)}(1),\cdots,f_{\mathrm{R}}^{(p)}(L_f-1)]^{\mathrm{T}}\\ \boldsymbol{f}_{\mathrm{I}}^{(p)}(k)=[f_{\mathrm{I}}^{(p)}(0),f_{\mathrm{I}}^{(p)}(1),\cdots,f_{\mathrm{I}}^{(p)}(L_f-1)]^{\mathrm{T}}\end{cases} \tag{7.4.7}$$

经过正交小波变换后第 $p$ 路子信道均衡器的输入信号为

$$\boldsymbol{R}^{(p)}(k)=\boldsymbol{R}_{\mathrm{R}}^{(p)}(k)+\mathrm{j}\boldsymbol{R}_{\mathrm{I}}^{(p)}(k)=\boldsymbol{V}\boldsymbol{y}_{\mathrm{R}}^{(p)}(k)+\mathrm{j}\boldsymbol{V}\boldsymbol{y}_{\mathrm{I}}^{(p)}(k) \tag{7.4.8}$$

设

$$\boldsymbol{R}_{\mathrm{R}}^{(p)}(k)=[u_{\mathrm{R},1,0}^{(p)}(k),u_{\mathrm{R},1,1}^{(p)}(k),\cdots,u_{\mathrm{I},J,k_J}^{(p)}(k),s_{\mathrm{R},J,0}^{(p)}(k),s_{\mathrm{R},J,1}^{(p)}(k),\cdots,s_{\mathrm{R},J,k_J}^{(p)}(k)]^{\mathrm{T}} \tag{7.4.9}$$

$$\begin{cases}u_{\mathrm{R},j,m}^{(p)}(k)=\sum\limits_{l=0}^{L_f-1}y_{\mathrm{R}}^{(p)}(k-l)\varphi_{j,m}^{(p)}(l)\\ s_{\mathrm{R},J,m}^{(p)}(k)=\sum\limits_{l=0}^{L_f-1}y_{\mathrm{R}}^{(p)}(k-l)\phi_{J,m}^{(p)}(l)\end{cases} \tag{7.4.10}$$

$$\boldsymbol{R}_{\mathrm{I}}^{(p)}(k)=[u_{\mathrm{I},1,0}^{(p)}(k),u_{\mathrm{I},1,1}^{(p)}(k),\cdots,u_{\mathrm{I},J,k_J}^{(p)}(k),s_{\mathrm{I},J,0}^{(p)}(k),s_{\mathrm{I},J,1}^{(p)}(k),\cdots,s_{\mathrm{I},J,k_J}^{(p)}(k)]^{\mathrm{T}} \tag{7.4.11}$$

$$\begin{cases}u_{\mathrm{I},j,m}^{(p)}(k)=\sum\limits_{l=0}^{L_f-1}y_{\mathrm{I}}^{(p)}(k-l)\varphi_{j,m}^{(p)}(p)\\ s_{\mathrm{I},J,m}^{(p)}(k)=\sum\limits_{l=0}^{L_f-1}y_{\mathrm{I}}^{(p)}(k-l)\phi_{J,m}^{(p)}(p)\end{cases} \tag{7.4.12}$$

第 $p$ 路子信道均衡器的输出信号 $z^{(p)}(k)$为

$$z^{(p)}(k)=z_{\mathrm{R}}^{(p)}(k)+\mathrm{j}z_{\mathrm{I}}^{(p)}(k)=\boldsymbol{f}_{\mathrm{R}}^{(p)\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{R}}^{(p)}(k)+\mathrm{j}\boldsymbol{f}_{\mathrm{I}}^{(p)\mathrm{T}}(k)\boldsymbol{R}_{\mathrm{I}}^{(p)}(k) \tag{7.4.13}$$

式中，$z_{\mathrm{R}}^{(p)}(k)$和$z_{\mathrm{I}}^{(p)}(k)$为第$p$路子信道均衡器输出信号的实部和虚部

$$\begin{cases} z_{\mathrm{R}}^{(p)}(k)=\boldsymbol{f}_{\mathrm{R}}^{\mathrm{T}(p)}(k)\boldsymbol{R}_{\mathrm{R}}^{(p)}(k) \\ z_{\mathrm{I}}^{(p)}(k)=\boldsymbol{f}_{\mathrm{I}}^{\mathrm{T}(p)}(k)\boldsymbol{R}_{\mathrm{I}}^{(p)}(k) \end{cases} \tag{7.4.14}$$

均衡器的输出为

$$z(k)=\sum_{p=0}^{P-1}z^{(p)}(k)=z_{\mathrm{R}}(k)+\mathrm{j}z_{\mathrm{I}}(k) \tag{7.4.15}$$

式中，$z_{\mathrm{R}}(k)$和$z_{\mathrm{I}}(k)$分别为均衡器输出信号的实部和虚部。

第$p$路子均衡器权向量的迭代公式为

$$\begin{cases} \boldsymbol{f}_{\mathrm{R}}^{(p)}(k+1)=\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)-\mu\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1(p)}(k)z_{\mathrm{R}}(k)e_{\mathrm{R}}(k)\boldsymbol{R}_{\mathrm{R}}^{(p)}(k) \\ \boldsymbol{f}_{\mathrm{I}}^{(p)}(k+1)=\boldsymbol{f}_{\mathrm{I}}^{(p)}(k)-\mu\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1(p)}(k)z_{\mathrm{I}}(k)e_{\mathrm{I}}(k)\boldsymbol{R}_{\mathrm{I}}^{(p)}(k) \end{cases} \tag{7.4.16}$$

式中，$e_{\mathrm{R}}(k)$和$e_{\mathrm{I}}(k)$分别表示误差函数$e(k)$的实部和虚部

$$\begin{cases} e_{\mathrm{R}}(k)=z_{\mathrm{R}}^{2}(k)-R_{\mathrm{R}}^{2} \\ e_{\mathrm{I}}(k)=z_{\mathrm{I}}^{2}(k)-R_{\mathrm{I}}^{2} \end{cases} \tag{7.4.17}$$

式中，$R_{\mathrm{R}}^{2}$和$R_{\mathrm{I}}^{2}$分别代表复信源实部和虚部的统计模值[67~70]。

$$\begin{cases} R_{\mathrm{R}}^{2}=E[|a_{\mathrm{R}}(k)|^{4}]/E[|a_{\mathrm{R}}(k)|^{2}] \\ R_{\mathrm{I}}^{2}=E[|a_{\mathrm{I}}(k)|^{4}]/E[|a_{\mathrm{I}}(k)|^{2}] \end{cases} \tag{7.4.18}$$

式中，$\hat{\boldsymbol{R}}_{\mathrm{R}}^{-1(p)}(k)=\mathrm{diag}[\sigma_{\mathrm{R},j,0}^{2(p)}(k),\sigma_{\mathrm{R},j,1}^{2(p)}(k),\cdots,\sigma_{\mathrm{R},j,k_j-1}^{2(p)}(k),\sigma_{\mathrm{R},J+1,0}^{2}(k),\cdots,\sigma_{\mathrm{R},J+1,k_j-1}^{2(p)}(k)]$，$\hat{\boldsymbol{R}}_{\mathrm{I}}^{-1(p)}(k)=\mathrm{diag}[\sigma_{\mathrm{I},j,0}^{2(p)}(k),\sigma_{\mathrm{I},j,1}^{2(p)}(k),\cdots,\sigma_{\mathrm{I},j,k_j-1}^{2(p)}(k),\sigma_{\mathrm{I},J+1,0}^{2}(k),\cdots,\sigma_{\mathrm{I},J+1,k_j-1}^{2(p)}(k)]$，$\sigma_{\mathrm{R},j,k}^{2(p)}(k)$与$\sigma_{\mathrm{R},J+1,k_j}^{2(p)}(k)$分别是第$p$路小波系数的实部$u_{\mathrm{R}j,m}^{(p)}(k)$和尺度变换系数的实部$s_{\mathrm{R}J,m}^{(p)}(k)$的平均功率估计，$\sigma_{\mathrm{I},j,k}^{2(p)}(k)$与$\sigma_{\mathrm{I},J+1,k_j}^{2(p)}(k)$分别是第$p$路小波系数的虚部$u_{\mathrm{I},j,m}^{(p)}(k)$和尺度变换系数的虚部$s_{\mathrm{I},J,m}^{(p)}(k)$的平均功率估计。它们的递推公式为

$$\begin{cases} \hat{\sigma}_{\mathrm{R},j,m}^{2(p)}(k+1)=\beta_{\sigma}\hat{\sigma}_{\mathrm{R},j,m}^{2(p)}(k)+(1-\beta_{\sigma})|u_{\mathrm{R},j,m}^{(p)}(k)|^{2} \\ \hat{\sigma}_{\mathrm{R},J+1,m}^{2(p)}(k+1)=\beta_{\sigma}\hat{\sigma}_{\mathrm{R},J+1,m}^{2(p)}(k)+(1-\beta_{\sigma})|s_{\mathrm{R},J,m}^{(p)}(k)|^{2} \end{cases} \tag{7.4.19}$$

$$\begin{cases} \hat{\sigma}_{\mathrm{I},j,m}^{2(p)}(k+1)=\beta_{\sigma}\hat{\sigma}_{\mathrm{I},j,m}^{2(p)}(k)+(1-\beta_{\sigma})|u_{\mathrm{I},j,m}^{(p)}(k)|^{2} \\ \hat{\sigma}_{\mathrm{I},J+1,m}^{2(p)}(k+1)=\beta_{\sigma}\hat{\sigma}_{\mathrm{I},J+1,m}^{2(p)}(k)+(1-\beta_{\sigma})|s_{\mathrm{I},J,m}^{(p)}(k)|^{2} \end{cases} \tag{7.4.20}$$

式中，$\beta_\sigma$ 为平滑因子，且 $0<\beta_\sigma<1$，一般 $\beta_\sigma$ 的取值比较接近于 1。

MTSDNAGA-T/P-FSE-WTMMA 的操作流程如图 7.21 所示。操作步骤如下。

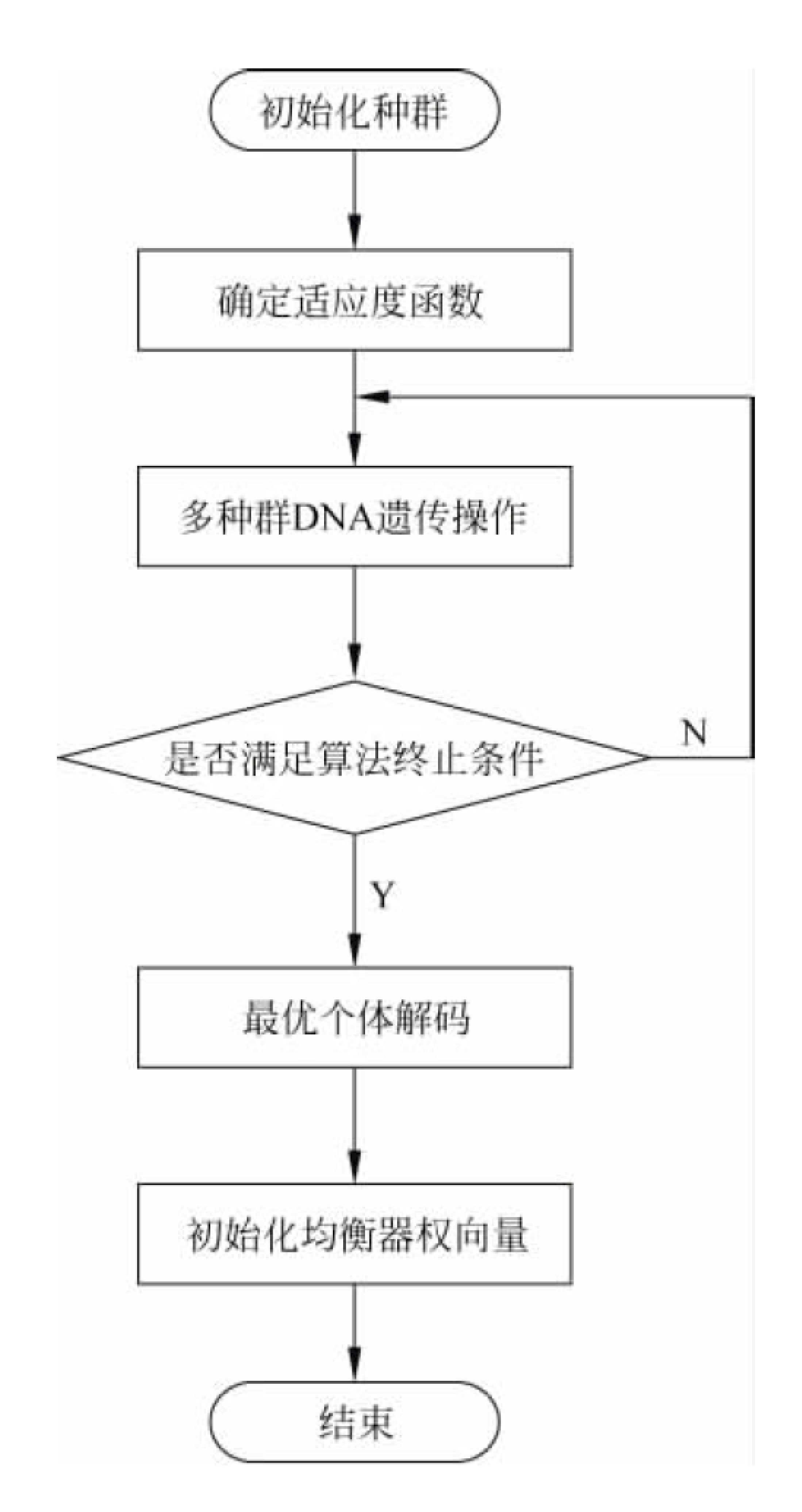



图 7.21 MTSDNAGA-T/P-WTMMA 操作流程

步骤 1：初始化种群。在 DNA 遗传算法中，每个 DNA 链个体对应于每个子均衡器的权向量，分别对均衡器权向量的实部和虚部进行优化。按照 7.1.1 节的编码方式将均衡器权向量编码为 DNA 核苷酸单链，设置种群规模为 $M$ 个个体。对于种群中的单链个体再生成与之配对的互补链，形成双链 DNA 种群。将 $\boldsymbol{y}^{(p)}(k)$作为 DNA 遗传算法的输入信号。将均衡器权向量用 DNA 遗传算法编码表示成 DNA 核苷酸序列，设初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_M]$。

步骤 2：确定适应度函数。由 WTMMA 的代价函数确定 DNA 遗传算法的适应度函数，定义为

$$\mathrm{Fit}(\boldsymbol{f}_m)=\frac{b}{J_{\mathrm{WTMMA}}(\boldsymbol{f}_m)} \tag{7.4.21}$$

式中，$J_{\mathrm{WTMMA}}=(|z_{\mathrm{R}}(k)|^2-R_{\mathrm{R}}^2)^2+(|z_{\mathrm{I}}(k)|^2-R_{\mathrm{I}}^2)^2$，$b$ 表示比例系数。

步骤 3：按照图 7.19 的多种群 DNA 遗传操作流程对种群中的个体进行操作。

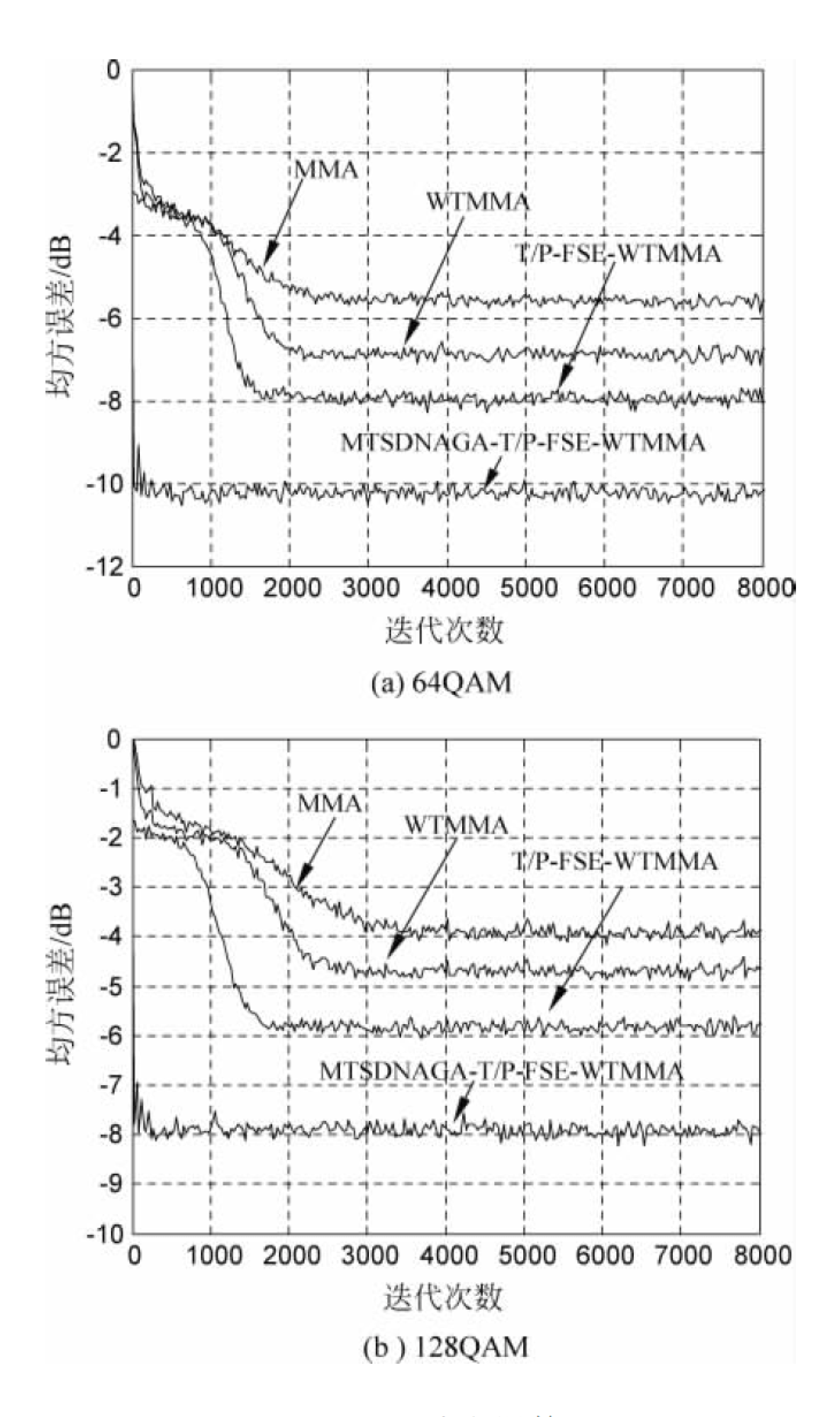
步骤 4：判断是否达到进化终止条件。如果当前进化代数小于最大进化代数，则继续对种群中的个体进行进化操作，直到满足进化终止条件为止；如果当前进化代数大于最大进化代数，则输出适应度值最大的单链 DNA 个体，并将该个体解码后作为均衡器初始优化权向量。

### 7.4.3 算法仿真

**【实验 7.4】** 信道 $\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$，发射信号为 64QAM，信噪比为 25dB，均衡器权长为 16，第 4 个抽头系数设置为 1，其余为 0，步长 $\mu_{\mathrm{MMA}}=0.000\,035$，$\mu_{\mathrm{WTMMA}}=0.000\,018$，$\mu_{\mathrm{T/P\text{-}FSE\text{-}WTMMA}}=0.000\,02$，$\mu_{\mathrm{MTSDNAGA\text{-}T/P\text{-}FSE\text{-}WTMMA}}=0.000\,028$；对信道的输入信号采用 DB2 正交小波分解，功率初始值设置为 4，遗忘因子 $\beta=0.99$。MTSDNAGA-T/P-FSE-WTMMA 的种群规模取 60，终止进化代数为 100。

信道 $\boldsymbol{h}=[0.9656\ -0.0906\ 0.0578\ 0.2368]$时，发射信号为 128QAM 调制信号，信噪比为 25dB，均衡器权长为 16，第 4 个抽头系数设置为 1；步长 $\mu_{\mathrm{MMA}}=0.47\times10^{-6}$，$\mu_{\mathrm{WTMMA}}=0.42\times10^{-6}$，$\mu_{\mathrm{T/P\text{-}FSE\text{-}WTMMA}}=0.3\times10^{-6}$，$\mu_{\mathrm{MTSDNAGA\text{-}T/P\text{-}FSE\text{-}WTMMA}}=0.37\times10^{-6}$；对信道的输入信号采用 DB2 正交小波分解，功率初始值设置为 4，遗忘因子 $\beta=0.99$。MTSDNAGA-T/P-FSE-WTMMA 的种群规模取 60，终止进化代数为 100。

在保证眼图完全清晰睁开的前提下，500 次蒙特卡罗仿真结果如图 7.22 所示。



(a) 64QAM

(b) 128QAM

图 7.22 均方误差

图 7.22(a)表明,MTSDNAGA-T/P-FSE-WTMMA 的收敛速度最快,比 MMA 快约 2300 步,比 WTMMA 快约 1800 步,比 T/P-FSE-WTMMA 快约 1400 步;在稳态误差上,MTSDNAGA-T/P-FSE-WTMMA 的稳态误差比 T/P-FSE-WTMMA 的稳态误差小约 2dB,比 WTMMA 的稳态误差小约 3dB,比 MMA 的稳态误差小约 4.5dB。图 7.22(b)表明,MTSDNAGA-T/P-FSE-WTMMA 的收敛速度最快,比 MMA 快约 3000 步,比 WTMMA 快约 2500 步,比 T/P-FSE-WTMMA 快约 1500 步;在稳态误差上,MTSDNAGA-T/

P-FSE-WTMMA 的稳态误差比 T/P-FSE-WTMMA 的稳态误差小约 1.5dB,比 WTMMA 的稳态误差小约 2dB,比 MMA 的稳态误差小约 3dB。

## 参考文献


[1] 丁永生,任立红,邵世煌. DNA 计算与软计算[M]. 北京:科学出版社,2002.

[2] 陈智华. 基于 DNA 计算自组装模型的若干密码问题研究[D]. 武汉:华中科技大学,2009.

[3] Lipton R. DNA solution of hard computation problem[J]. Science, 1995, 268(4):542-545.

[4] Roweis S, Winfree E, Burgoyne R, et al. A sticker-based model for DNA computation[J]. Journal of Computational Biology,1998,5(4):615-629.

[5] Ouyang Q, Kaplan P D, Liu S, et al. DNA solution of the maximal clique problem[J]. Science,1997,278(17):446-449.

[6] Liu Q H, Frutos A G, Wang L, et al. Progress toward demonstration of a surface based DNA computation: A one word approach to solve a model satisfiability Problem[J]. Biosystems,1999,52(1-3):25-33.

[7] Liu Q H, Wang L M, Anthony G, et al. DNA computing on surfaces[J]. Nature,2000,403:175-179.

[8] Wu H Y. An improved surface-based method for DNA computation[J]. Biosystems,2001,59(1):1-5.

[9] 陈宵. DNA 遗传算法及应用研究[D]. 杭州:浙江大学,2010.

[10] 戴侃. DNA 遗传算法及在化工过程中的应用[D]. 杭州:浙江大学,2010.

[11] Kwasnicka H, Przewozniczek M. Multi population pattern searching algorithm: a new evolutionary method based on the idea of messy genetic algorithm[J]. IEEE Transactions on Evolutionary Computation,2011,15(5):715-734.

[12] Lei Yang, Shiqing Li, Nan Xiong, et al. Industrial enterprise distribution network planning based on multiple population genetic algorithm[C]. 2014 International Conference on Power System Technology,2014:550-555.

[13] Zhang D, Xia Y L, He X X, et al. Multi-step evolution strategy based DNA genetic algorithm for parameters estimating[C]. The 4th International Conference

on Intelligent Control and Information Processing(ICICIP). Beijing,2013：828-835.

[14] 张冰龙.基于自适应双链 DNA 遗传优化的盲均衡算法[D].南京：南京信息工程大学,2015.

[15] 郭业才,张冰龙,吴彬彬.基于 DNA 遗传优化的正交小波常模盲均衡算法[J].数据采集与处理,2014,29(3)：366-371.

[16] Federico L, Brunilde S. A Tabu search algorithm for the location of data centers and software components in green cloud computing networks[J]. IEEE Transactions on Cloud Computing. 2013,1(1)：22-35.

[17] Xiangning Lin, Shuohao Ke,Zhengtian Li,et al. A fault diagnosis method of power systems based on improved objective function and genetic Algorithm-Tabu search[J]. IEEE Transactions on Power Delivery,2010,25(3)：1268-1274.

[18] 李刚,刘景法.基于禁忌搜索的启发式算法求解带平衡约束的圆形体安装问题[J].中国科学：信息科学,2011,41(9)：1076-1088.

[19] 张昊,陶然,李志勇等.基于 KNN 算法及禁忌搜索算法的特征选择方法在入侵检测中的应用研究[J].电子学报,2009,37(7)：1628-1932.

[20] Carcangiu S, Fanni A,Montisci A. Multiobjective tabu search algorithm for optimal design of electromagnetic devices[J]. IEEE Transactions on Magnetics, 2008,44(6)：970-973.

[21] 马华伟,杨善林.可选时间窗车辆调度问题的改进禁忌搜索算法[J].系统仿真学报,2008,20(16)：4454-4457.

[22] Guo Yecai, Zhang Binglong. A New DNA Algorithm for solving the minimum set covering problem based on molecular beacon[J]. Advances in Communication Technology and Systems,2014,56：361-369.

[23] Ramirez-Rosado I J, Dominguez-Navarro J A. New multiobjective Tabu search algorithm for fuzzy optimal planning of power distribution systems[J]. IEEE Transactions on Power Systems,2006,21(1)：224-233.

[24] 郭业才,张洁茹,张冰龙.基于禁忌搜索的双链 DNA 计算小波盲均衡算法[J].系统仿真学报,2017,29(1)：21-26.

[25] Youcong Ni, Xin Du,Ruliang Xiao,et al. Multi-modulus blind equalization algorithm based on high-order QAM genetic optimization[C]. 2012 Eighth International Conference on Natural Computation,2012：679-682.

[26] Junwen Zhang, Jianjun Yu,Nan Chi,et al. Multi-modulus blind equalizations for coherent quadrature duobinary spectrum shaped PM-QPSK digital signal

processing[J]. Journal of Lightwave Technology,2013,31(7):1073-1078.

[27] 杨俊.基于多种群遗传算法的边坡稳定性分析方法研究[D].兰州:兰州理工大学,2014.

[28] Yukang Tian, Huy Quan Vu. A multiple population genetic algorithm and its application in fuzzy controller[C]. 2010 Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI), 2010: 729-733.

[29] Tan W S, Hassan M Y, Majid M S. Multi-population genetic algorithm for allocation and sizing of distributed generation[C]. 2012 IEEE International Power Engineering and Optimization Conference(PEDCO),2012: 108-113.

[30] Li Y M, Li W, Yan W. Daily generation scheduling for reducing unit regulating frequency using multi-population genetic algorithm[C]. 2012 IEEE Power and Energy Society General Meeting,2012: 1-7.

[31] 李军华.基于知识和多种群进化的遗传算法研究[D].南京:南京航空航天大学,2009.

# 第 8 章　基于 DNA 智能群算法优化的盲均衡算法

**【内容导读】** 本章将 DNA 遗传算法与人工鱼群算法相结合，得到了 DNA 人工鱼群算法，研究了基于 DNA 人工鱼群算法优化的多模盲均衡算法（包括基于人工鱼群算法优化 DNA 序列的多模盲均衡算法、基于 DNA 遗传人工鱼群优化 DNA 序列的加权多模盲均衡算法）；将 DNA 遗传算法、混沌算法和人工鱼群算法相结合，研究了基于 DNA 混沌人工鱼群优化 DNA 序列的频域加权多模盲均衡算法；将 DNA 遗传算法和蛙跳算法相结合，提出了 DNA 遗传蛙跳算法概念，研究了基于 DNA 遗传蛙跳算法优化的盲均衡算法；将 DNA 遗传算法和蝙蝠算法相结合，提出了 DNA 遗传蝙蝠算法概念，研究了基于 DNA 遗传蝙蝠算法的分数间隔多模盲均衡算法。

## 8.1 基于 DNA 人工鱼群算法优化的多模盲均衡算法

人工鱼群算法[1-8]具有快速寻优的特点，但其收敛后期易陷入局部搜索，同时，DNA 编码序列通过汉明约束条件保持了种群的多样性，使其不易陷入局部极值[9-19]。因此，本节通过汉明约束条件约束人工鱼群算法，提高其全局搜索能力，再与盲均衡算法[20-27]结合，研究了一类基于人工鱼群算法优化 DNA 序列的多模盲均衡算法[28-35]，通过优化均衡器初始权向量提高了盲均衡算法性能。

### 8.1.1 基于人工鱼群算法优化 DNA 序列的多模盲均衡算法

#### 1. 算法原理

针对 CMA 均衡多模信号时存在收敛速度慢、剩余均方误差大的缺陷，采用 MMA 和基于 DNA 核苷酸序列的编码方式表示问题的可能解，利用人工鱼群算法收敛速度快和搜索能力强的优点，引入 DNA 约束模型和代价函数，通过两个适应度函数寻找 DNA 编码序列中的最优序列，将最优序列解码后分别作为多模盲均衡算法最优初始权向量的实部和虚部，得到了一种基于人工鱼群算法优化 DNA 序列的多模盲均衡算法（artificial fish swarm intelligent optimization of DNA sequences based MMA，AFS-DNA-MMA），其原理图如图 8.1 所示。

图 8.1 中，$\boldsymbol{a}(k)$是零均值独立同分布的发射信号向量；$\boldsymbol{h}(k)$是信道脉冲响应向量；$\boldsymbol{w}(k)$是加性高斯白噪声向量；$\boldsymbol{y}(k)$是均衡器的接收信号向量，$\boldsymbol{y}_{\mathrm{R}}(k)$和$\boldsymbol{y}_{\mathrm{I}}(k)$是$\boldsymbol{y}(k)$的实部和虚部；$\boldsymbol{f}_{\mathrm{R}}(k)$和$\boldsymbol{f}_{\mathrm{I}}(k)$是均衡器权向量$\boldsymbol{f}(k)$的实部和虚部；$z(k)$是盲均衡器的输出信号，$z_{\mathrm{R}}(k)$和$z_{\mathrm{I}}(k)$是$z(k)$的实部和虚部；$e_{\mathrm{R}}(k)$和$e_{\mathrm{I}}(k)$是误差函数$e(k)$的实部和虚部；$\Psi_{\mathrm{R}}(\cdot)$与$\Psi_{\mathrm{I}}(\cdot)$分别是误差$e_{\mathrm{R}}(k)$和$e_{\mathrm{I}}(k)$的生成函数。

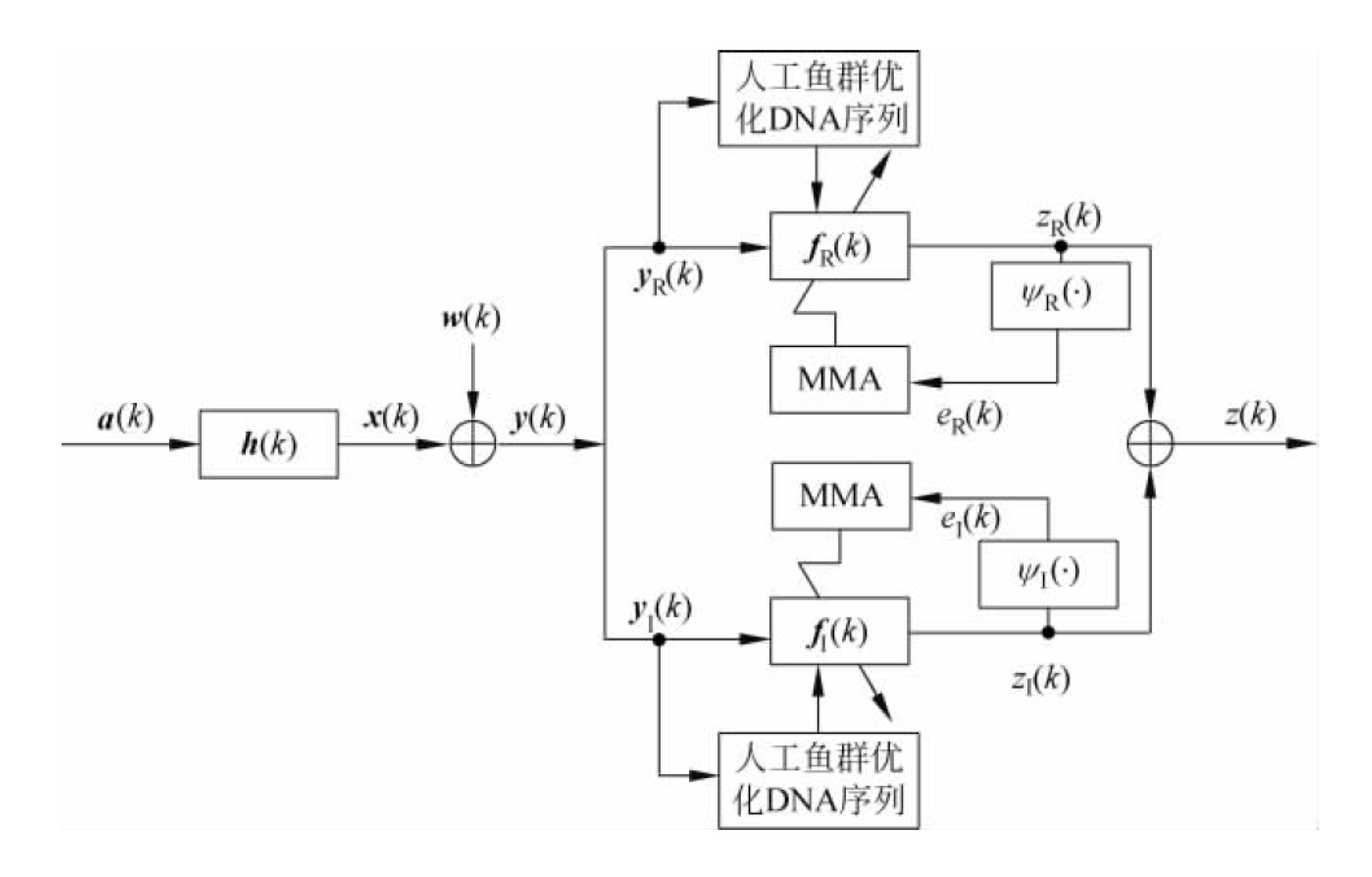



图 8.1 AFS-DNA-MMA 原理

由图 8.1 可知，MMA 输出信号 $z(k)$的实部和虚部分别为

$$\begin{cases} z_R(k) = \boldsymbol{f}_R^T(k)\boldsymbol{y}_R(k) \\ z_I(k) = \boldsymbol{f}_I^T(k)\boldsymbol{y}_I(k) \end{cases} \tag{8.1.1}$$

MMA 的误差函数 $e(k)$的实部和虚部分别为

$$\begin{cases} e_R(k) = z_R^2(k) - R_R^2 \\ e_I(k) = z_I^2(k) - R_I^2 \end{cases} \tag{8.1.2}$$

式中，$R_R^2$ 和 $R_I^2$ 分别代表发射信号实部和虚部的统计模值。

$$\begin{cases} R_R^2 = E[|a_R(k)|^4]/E[|a_R(k)|^2] \\ R_I^2 = E[|a_I(k)|^4]/E[|a_I(k)|^2] \end{cases} \tag{8.1.3}$$

因此，MMA 的代价函数为

$$J_{MMA} = E[e_R^2(k)] + E[e_I^2(k)] \tag{8.1.4}$$

MMA 权向量的迭代公式为

$$\begin{cases} \boldsymbol{f}_R(k+1) = \boldsymbol{f}_R(k) - \mu e_R(k)\boldsymbol{y}_R(k)z_R(k) \\ \boldsymbol{f}_I(k+1) = \boldsymbol{f}_I(k) - \mu e_I(k)\boldsymbol{y}_I(k)z_I(k) \end{cases} \tag{8.1.5}$$

式中，$\mu$ 为迭代步长。

2. 优化步骤

人工鱼群算法优化DNA序列初始化权向量流程如图8.2所示。

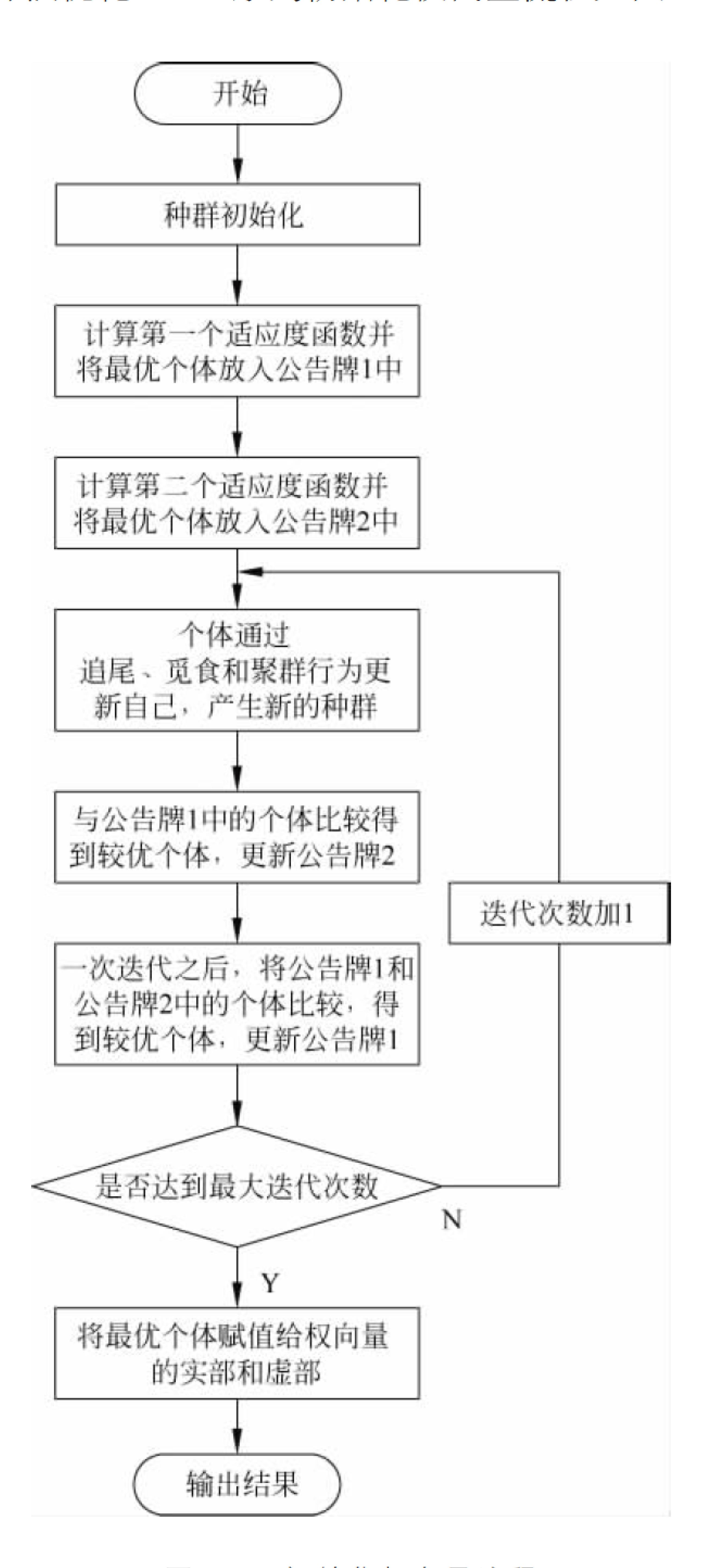



图8.2　初始化权向量流程

步骤 1：种群初始化并解码。

设 DNA 编码序列的初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N]$，其中 $\boldsymbol{f}_i$ 对应于 DNA 编码序列中的第 $i$ 个 DNA 编码序列，$1<i<N$，$N$ 是 DNA 编码序列的个数；再将 DNA 编码序列按式(7.1.2)和式(7.1.3)进行解码，得到十进制表示位置向量的种群。

步骤 2：确定两个适应度函数。

盲均衡算法的目的是得到均衡器最优权向量，即代价函数处于最小值状态，而人工鱼群算法的目的是寻找食物浓度值最大时所对应的人工鱼个体的位置向量。

因此，第一个适应度函数定义为代价函数的倒数，即

$$Y_1(\boldsymbol{f}_i)=\frac{1}{J(\boldsymbol{f}_i)},\quad i=1,2,\cdots,N \tag{8.1.6}$$

式中，$J(\boldsymbol{f}_i)=J_{\mathrm{MMA}}$为盲均衡器的代价函数，$\boldsymbol{f}_i$ 为第 $i$ 条人工鱼的位置向量。

第二个适应度函数是采用加权平均值法来处理汉明距离约束项的函数，即

$$Y_2(\boldsymbol{f}_i)=w_1\cdot H_1(\boldsymbol{f}_i,\boldsymbol{f}_i) \tag{8.1.7}$$

式中，$\boldsymbol{f}_i$ 对应于 DNA 编码序列中的第 $i$ 个 DNA 序列，为计算方便，设置 $w_1=1$。

步骤 3：计算两个适应度函数。

按式(8.1.6)计算人工鱼群算法的位置向量的第一个适应度值，并将第一个适应度最大值及其对应的十进制位置向量记录在公告牌 1 中；再按式(8.1.7)计算 DNA 编码序列位置向量的第二个适应度值，并将第二个适应度最大值及其对应的 DNA 编码序列的位置向量记录在公告牌 2 中。

步骤 4：执行人工鱼群算法并更新公告牌 2。

按十进制位置向量进行人工鱼群算法中的觅食、聚群、追尾等行为，并将第二个适应度函数式(8.1.7)作为适应度函数，将十进制位置向量进行反解码，得到四进制 DNA 编码序列，并计算其第二个适应度值，与公告牌 2 中保存的原第二个适应度最大值进行比较，如果第二个适应度值大，则用第

二个适应度值及其对应的四进制 DNA 编码序列更新公告牌 2 原先保存的内容。

步骤 5：更新公告牌 1。

在一次迭代之后，将公告牌 2 中的 DNA 序列进行解码，再利用第一个适应度函数式(8.1.6)计算其第一个适应度值，与公告牌 1 中保存的原第一个适应度最大值进行比较，如果第一个适应度值大，则用第一个适应度值及其对应的十进制位置向量更新公告牌 1 原先保存的内容。

步骤 6：判断是否满足结束条件。

如果当前迭代次数已经达到设定值，则将公告牌 1 中十进制位置向量作为最优位置向量输出，并将其作为多模盲均衡器最优初始权向量的实部和虚部；若不满足结束条件，则返回步骤 4。

### 3. 算法仿真

为了检验 AFS-DNA-MMA 的性能，以 CMA 和 MMA 为对比对象，进行仿真实验。

CMA、MMA 的均衡器权长为 11，第 6 个抽头系数设置为 1，其余为 0，信噪比 $SNR$＝20dB，样本个数 $N$＝10 000。

AFS-DNA-MMA 的种群规模取 30，人工鱼群中拥挤度因子取 0.68，人工鱼步长取 0.2，视野取 0.8，人工鱼群迭代次数为 20。

**【实验 8.1】** 当发射信号 8PSK 时，信道噪声为高斯白噪声，步长 $\mu_{\text{CMA}}=0.002$、$\mu_{\text{MMA}}=0.002$，$\mu_{\text{AFS-DNA-MMA}}=0.003$。当发射信号 16QAM 时，步长 $\mu_{\text{CMA}}=0.0001$、$\mu_{\text{MMA}}=0.0001$，$\mu_{\text{AFS-DNA-MMA}}=0.0002$。在保证眼图完全清晰睁开的前提下，蒙特卡罗仿真结果如图 8.3 所示。

图 8.3(a)表明，在收敛速度上，与 CMA、MMA 的收敛速度相比，AFS-DNA-MMA 比 CMA 快大约 2000 步，比 MMA 快大约 1500 步，收敛速度有了明显的提高；在均方误差上，AFS-DNA-MMA 与 CMA 相比，减少了 7.2dB，与 MMA 相比，减少了 5dB。图 8.3(b)表明，从收敛速度上看，与 CMA、MMA 的收敛速度相比，AFS-DNA-MMA 比 CMA 快大约 3500 步，

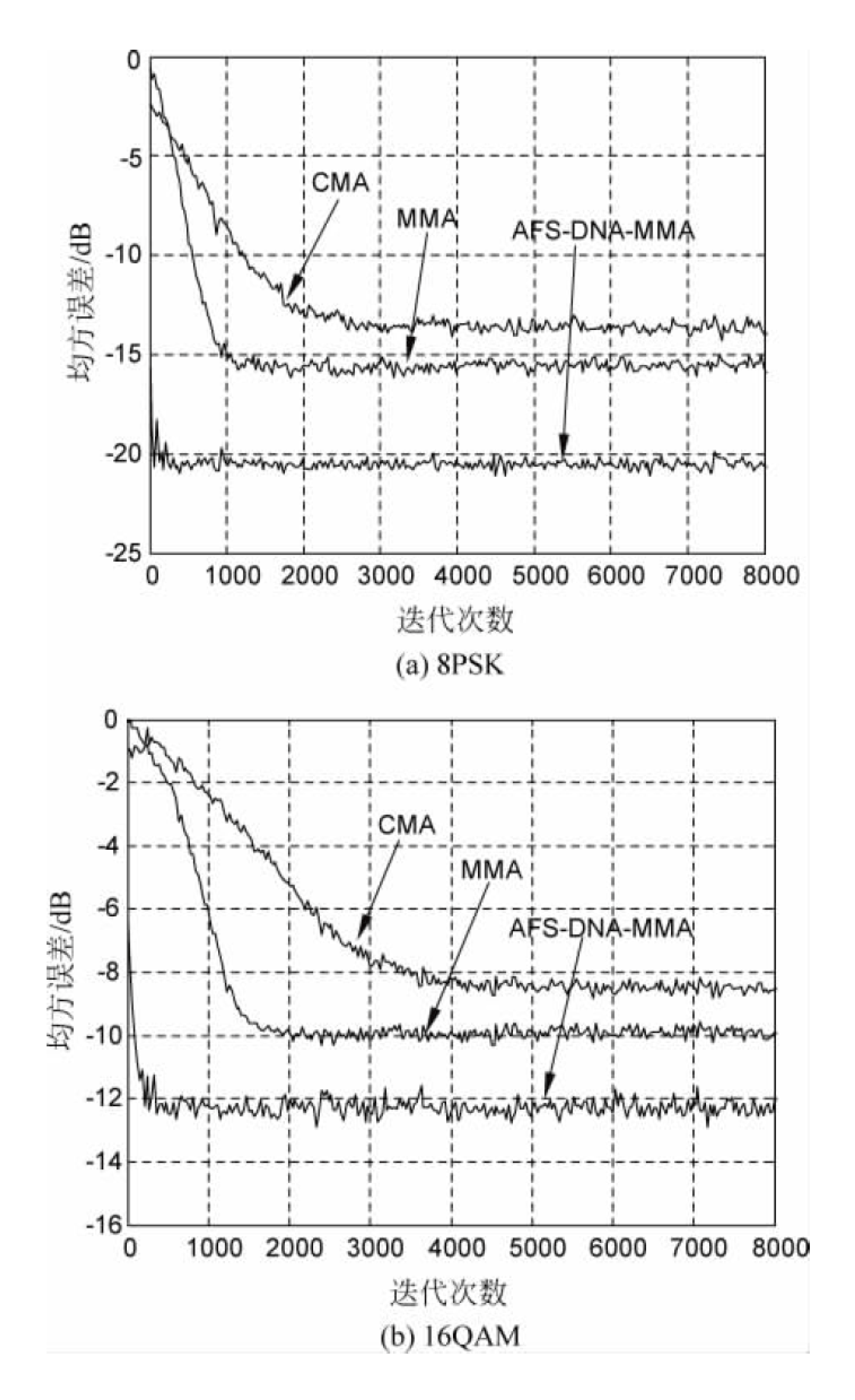



图 8.3 均方误差

比 MMA 快大约 1000 步,收敛速度有了明显的提高;从均方误差上看,AFS-DNA-MMA 与 CMA 相比,减少了 4dB,与 MMA 相比,减少了 2.1dB。

### 8.1.2 基于 DNA 遗传人工鱼群优化 DNA 序列的加权多模盲均衡算法

DNA 遗传算法的思想与遗传算法相似,但其采用四进制编码方式,提高了种群多样性及全局搜索能力。本节将 DNA 遗传算法和人工鱼群算法

相结合，定义了DNA遗传人工鱼群算法，并研究了一种基于DNA遗传人工鱼群算法优化DNA序列的加权多模盲均衡算法。

1. DNA遗传人工鱼群算法

人工鱼群算法收敛速度快、寻优精度高，但易于陷入局部收敛；而DNA遗传算法采用DNA编码方式，利用交叉、变异和倒位操作优化DNA序列，具有强大的全局搜索能力。将DNA遗传算法和人工鱼群算法相结合，得到的DNA遗传人工鱼群算法（DNA genetic artificial fish swarm algorithm，DNA-G-AFSA）兼具了两者的优点，弥补了人工鱼群算法的缺陷。

DNA遗传人工鱼群算法流程，如图8.4所示。

DNA遗传人工鱼群算法的实施步骤如下。

步骤1：人工鱼群种群初始化。

设人工鱼群初始种群 $\boldsymbol{f}_{\mathrm{AFS}}=(\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N)$，其中 $\boldsymbol{f}_i(i=1,2,\cdots,N)$ 为第 $i$ 条人工鱼的初始位置向量，$N$ 为人工鱼群中个体数量。

步骤2：计算适应度函数。

计算每条人工鱼的适应度值，将适应度最大值及其对应的位置向量分别记录在公告牌中。

步骤3：人工鱼群的行为选择。

人工鱼群中每条人工鱼发生追尾行为操作，若追尾不成功，则执行聚群行为操作；若聚群行为不成功，则执行觅食行为操作，人工鱼的当前位置向量发生改变。

步骤4：反解码操作。

将人工鱼群新的位置向量按式(7.1.2)和式(7.1.3)进行反解码，将十进制位置向量转换成四进制DNA编码序列，产生DNA编码序列种群。

步骤5：交叉操作。

从DNA编码序列种群中随机选择两个DNA编码序列作为父体，进行交叉操作，得到两个新DNA编码序列，并代替父体，得到新的DNA编码序

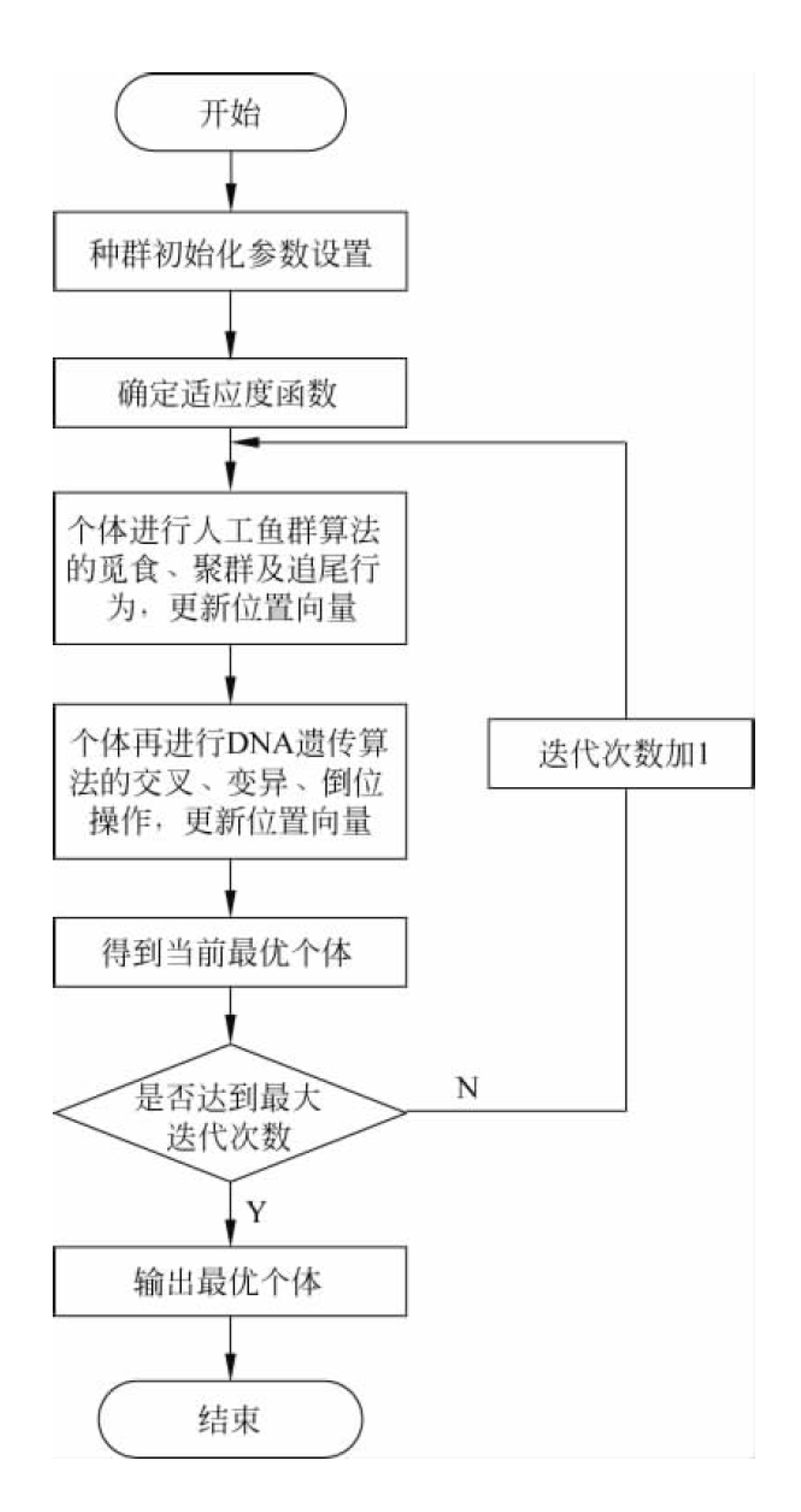



图 8.4　DNA 遗传人工鱼群算法流程

列种群。

步骤 6：变异操作。

再从 DNA 编码序列种群中随机选择一个 DNA 编码序列作为父体，随机产生一个(0,1)之间的随机数，进行变异操作，得到一个新 DNA 编码序列，并代替父体，得到新的 DNA 编码序列种群。

步骤 7：倒位操作。

随机产生一个(0,1)之间的随机数 $p_i$ 作为概率，以概率 $p_i$ 从 DNA 编

码序列种群中随机选择 DNA 编码序列作为父体，进行倒位操作，得到新 DNA 编码序列，并代替父体，得到新的 DNA 编码序列种群。

步骤 8：更新公告牌。

将 DNA 编码序列种群进行解码，得到十进制位置向量，即人工鱼群新的位置向量，计算每条人工鱼新位置向量对应的适应度值，将该适应度值与公告牌中保存的适应度最大值进行比较；如果当前适应度值大，则用当前适应度最大值及其对应的人工鱼位置向量更新公告牌中内容。

步骤 9：判断是否满足结束条件。

判断 DNA 遗传人工鱼群算法的迭代次数是否达到最大迭代次数，若满足条件，则输出公告牌中的位置向量，终止算法；若没有达到条件，跳转到步骤 3。

## 2. DNA 遗传人工鱼群 DNA 序列优化加权多模盲均衡算法

### (1) 加权多模盲均衡算法

加权多模盲均衡算法(WMMA)通过动态调整加权因子 $\lambda$ 的取值来降低均方误差，因此，如何选取 $\lambda$ 的值是保证加权多模盲均衡算法性能的关键，取值的大小依赖于大量仿真实验得出的数据，以确保 WMMA 的收敛速度和均方误差都优于 MMA。

WMMA 误差函数的实部和虚部定义为

$$e_{\mathrm{R}}(k) = z_{\mathrm{R}}(k)(z_{\mathrm{R}}^2(k) - |z_{\mathrm{R}}(k)|^{\lambda}R_{\mathrm{R}}^2) \tag{8.1.8}$$

$$e_{\mathrm{I}}(k) = z_{\mathrm{I}}(k)(z_{\mathrm{I}}^2(k) - |z_{\mathrm{I}}(k)|^{\lambda}R_{\mathrm{I}}^2) \tag{8.1.9}$$

式中，$R_{\mathrm{R}}$ 和 $R_{\mathrm{I}}$ 分别代表发射信号实部和虚部的统计模值，且

$$R_{\mathrm{R}}^2 = E[|a_{\mathrm{R}}(k)|^4]/E[|a_{\mathrm{R}}(k)|^{2+\lambda}] \tag{8.1.10}$$

$$R_{\mathrm{I}}^2 = E[|a_{\mathrm{I}}(k)|^4]/E[|a_{\mathrm{I}}(k)|^{2+\lambda}] \tag{8.1.11}$$

式中，$\lambda\in[0,2]$为加权因子。

WMMA 权向量的更新公式为

$$\boldsymbol{f}_{\mathrm{R,WMMA}}(k+1) = \boldsymbol{f}_{\mathrm{R,WMMA}}(k) - \mu e_{\mathrm{R}}(k)\boldsymbol{y}_{\mathrm{R}}^{*}(k) \tag{8.1.12}$$

$$\boldsymbol{f}_{\mathrm{I,WMMA}}(k+1) = \boldsymbol{f}_{\mathrm{I,WMMA}}(k) - \mu e_{\mathrm{I}}(k) \boldsymbol{y}_{\mathrm{I}}^{*}(k) \tag{8.1.13}$$

式中，$\boldsymbol{f}_{\mathrm{R,WMMA}}(k)$和 $\boldsymbol{f}_{\mathrm{I,WMMA}}(k)$分别为 WMMA 均衡器权向量的实部和虚部。

(2) DNA 遗传人工鱼群优化 DNA 序列的加权多模盲均衡算法

将 DNA 遗传人工鱼群优化 DNA 编码序列方法和 WMMA 相结合，研究了一种 DNA 遗传人工鱼群优化 DNA 序列的加权多模盲均衡算法(genetic artificial fish swarm optimization DNA encoding sequences based weighted multi-modulus algorithm, G-AFS-DNA-WMMA)，其原理如图 8.5 所示，实现步骤如下。

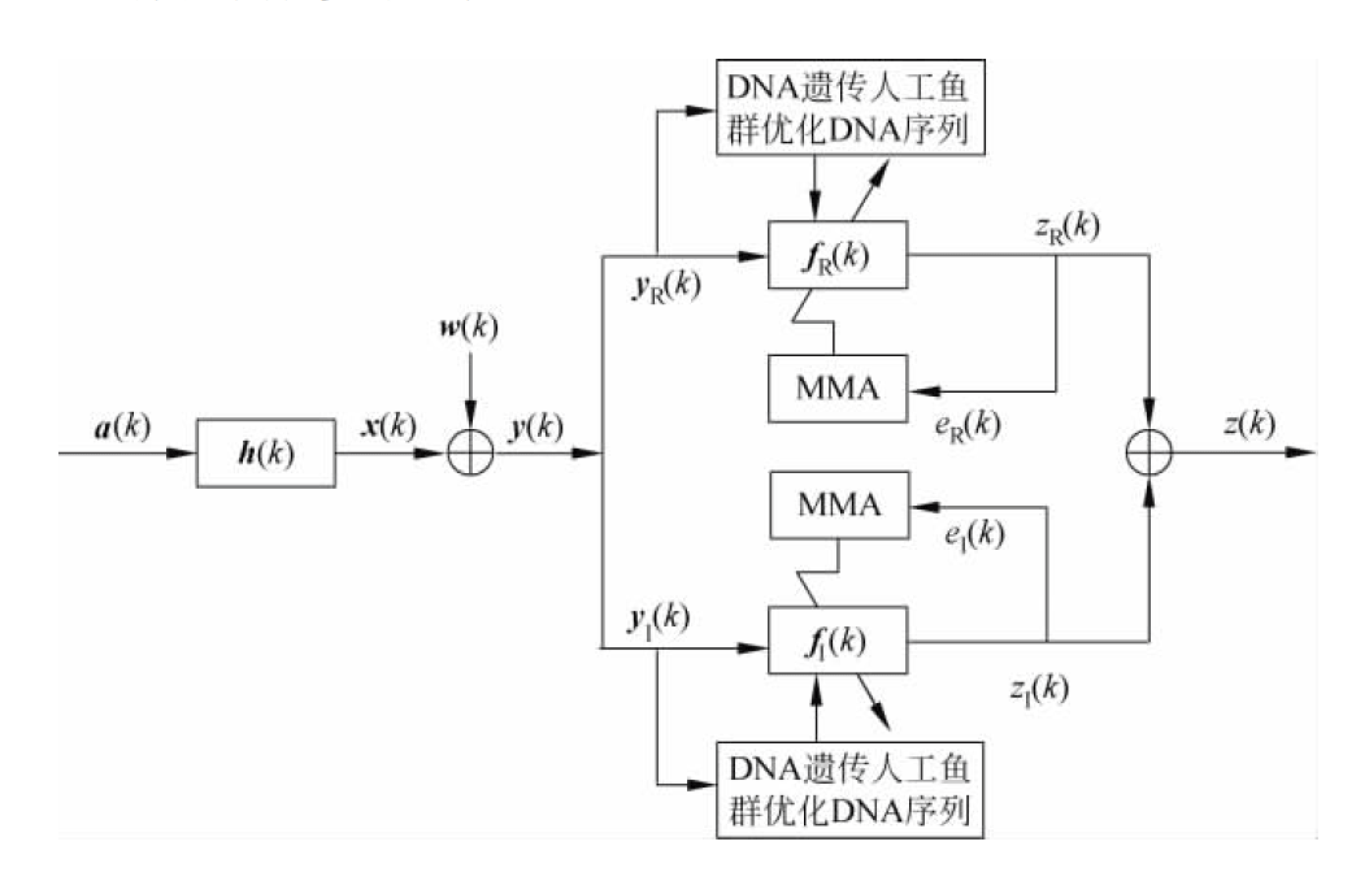



图 8.5 G-AFS-DNA-WMMA 原理

步骤 1：种群初始化并解码。

设 DNA 编码序列的初始种群 $\boldsymbol{f}=[\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N]$，其中 $\boldsymbol{f}_i$ 对应于 DNA 编码序列中的第 $i$ 个 DNA 编码序列，$1<i<N$，$N$ 是 DNA 编码序列的个数；再随机设置一组 DNA 编码序列 $\boldsymbol{f}_0$ 作为计算相似度的对比序列。将 DNA 编码序列按式(7.1.2)与式(7.1.3)进行解码，得到十进制位置向量种群。

步骤 2：确定两个适应度函数。

第一个适应度函数定义为代价函数的倒数，即

$$Y_1(\boldsymbol{f}_i)=\frac{1}{J(\boldsymbol{f}_i)},\quad i=1,2,\cdots,N \tag{8.1.14}$$

式中，$J(\boldsymbol{f}_i)=J_{\mathrm{WMMA}}(\boldsymbol{f}_i)$为盲均衡器的代价函数，$\boldsymbol{f}_i$ 为 DNA 遗传人工鱼群算法优化的第 $i$ 条人工鱼的位置向量。

第二个适应度函数采用加权平均值法来处理汉明距离和相似度约束项的函数，即

$$Y_2(\boldsymbol{f}_i)=p_1\cdot H_1(\boldsymbol{f}_i,\boldsymbol{f}_i)+p_2\cdot H_2(\boldsymbol{f}_i,\boldsymbol{f}_0) \tag{8.1.15}$$

式中，$\boldsymbol{f}_i$ 对应于 DNA 编码序列中的第 $i$ 个 DNA 序列，$\boldsymbol{f}_0$ 是计算相似度的对比序列。为计算方便，设置 $p_1=p_2=1$。

步骤 3：计算两个适应度函数。

按式(8.1.14)计算 DNA 遗传人工鱼群的位置向量对应的第一个适应度值，并将第一个适应度最大值及其对应的十进制位置向量记录在公告牌 1 中；再按式(8.1.15)计算 DNA 编码序列位置向量的第二个适应度值，并将第二个适应度最大值及其对应的 DNA 编码序列的位置向量记录在公告牌 2 中。

步骤 4：执行 DNA 遗传人工鱼群算法并更新公告牌 2。

将第二个适应度函数式(8.1.15)作为 DNA 遗传人工鱼群算法的适应度函数，按十进制位置向量进行人工鱼群算法中的觅食、聚群、追尾等行为，将十进制位置向量进行反解码，得到四进制 DNA 编码序列，进行 DNA 遗传算法的交叉、变异、倒位操作，并计算其第二个适应度值，与公告牌 2 中保存的原第二个适应度最大值进行比较，如果第二个适应度值大，则用第二个适应度值及其对应的四进制 DNA 编码序列更新公告牌 2 原先保存的内容。

步骤 5：更新公告牌 1。

在一次迭代之后，将公告牌 2 中的 DNA 序列进行解码，再利用第一个适应度函数式(8.1.14)，计算其第一个适应度值，与公告牌 1 中保存的原第

一个适应度最大值进行比较，如果第一个适应度值大，则用第一个适应度值及其对应的十进制位置向量更新公告牌1原先保存的内容。

步骤6：判断是否满足结束条件。

如果当前迭代次数已经达到设定值，则将公告牌1中十进制位置向量作为最优位置向量输出，并将其作为WMMA最优初始权向量的实部和虚部；反之，则返回步骤4。

（3）算法仿真

为了检验G-AFS-WMMA的性能，将MMA和WMMA作为对比对象，进行仿真实验。

MMA、WMMA的均衡器权长为11，信噪比$SNR$＝20dB，样本个数$N$＝10 000。

G-AFS-DNA-WMMA的种群规模取30，人工鱼群的拥挤度因子取0.68，人工鱼步长取0.2，视野取0.8，人工鱼群的迭代次数为20。混合相位水声信道$\boldsymbol{h}=[0.3132\quad -0.1040\quad 0.8908\quad 0.3134]$，其在单位圆内的零点有3个。

**【实验8.2】** 信道噪声为高斯白噪声，采用8PSK发射信号时，步长$\mu_{\text{MMA}}=0.002$，$\mu_{\text{WMMA}}=0.002$，$\mu_{\text{G-AFS-DNA-WMMA}}=0.005$；采用16QAM发射信号时，步长$\mu_{\text{MMA}}=0.000\,08$、$\mu_{\text{WMMA}}=0.000\,08$，$\mu_{\text{G-AFS-DNA-WMMA}}=0.0002$。在保证眼图完全清晰睁开的条件下，500次蒙特卡罗仿真结果如图8.6所示。

图8.6(a)知，从收敛速度上看，与MMA、WMMA的收敛速度相比，G-AFS-DNA-WMMA比MMA快大约4500步，比WMMA快大约900步，收敛速度有了明显的提高；从均方误差上看，G-AFS-DNA-WMMA与MMA相比，减少了3.8dB，与WMMA相比，减少了2.4dB。图8.6(b)可知，从收敛速度上看，与MMA、WMMA的收敛速度相比，G-AFS-DNA-WMMA比MMA快大约1000步，比WMMA快大约950步，收敛速度有了明显的提高；从均方误差上看，G-AFS-DNA-WMMA与MMA相比，减少了3.2dB，与WMMA相比，减少了0.4dB。

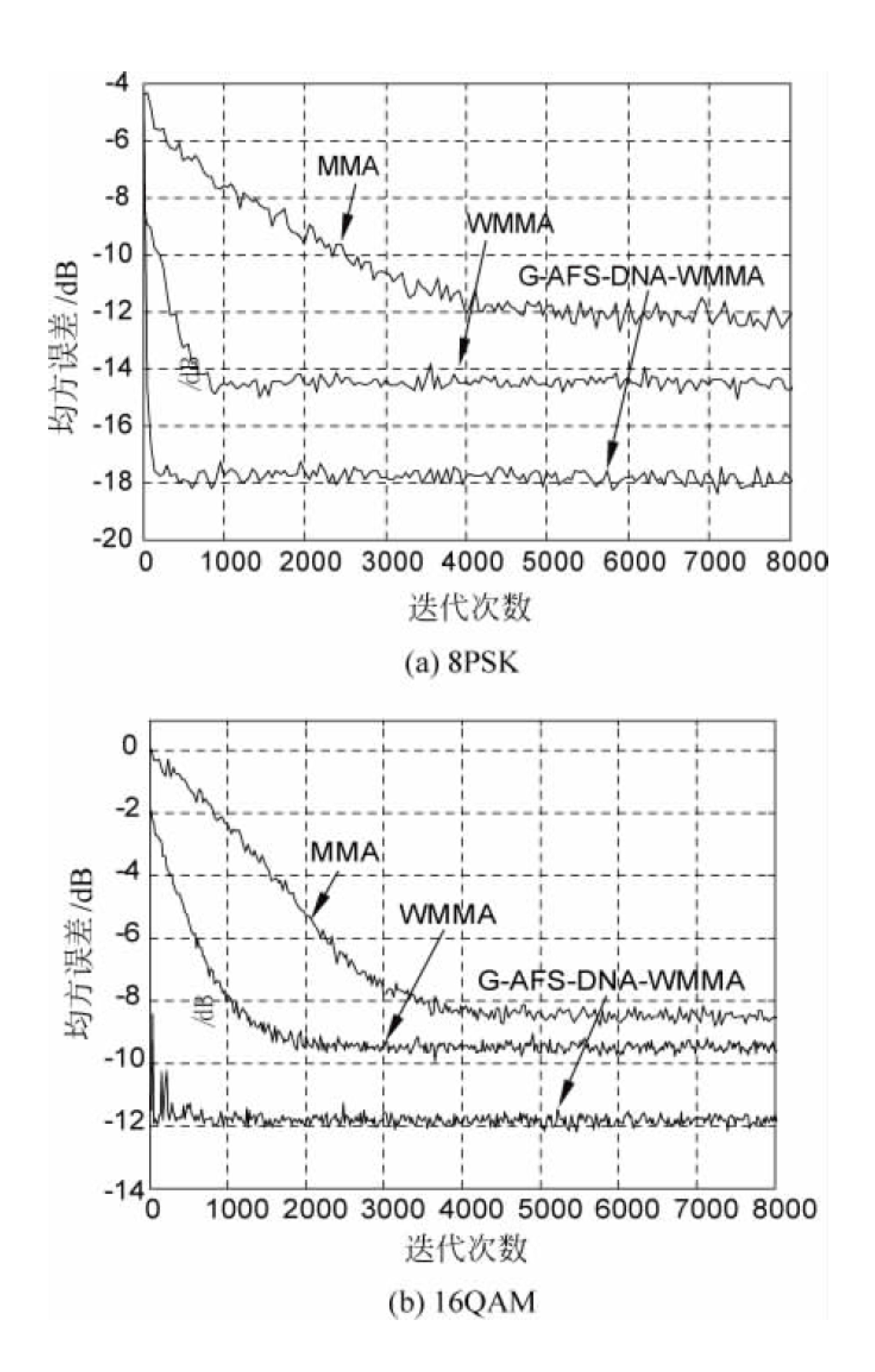



图 8.6　均方误差

### 8.1.3　基于 DNA 混沌人工鱼群算法优化 DNA 序列的频域加权多模盲均衡算法

DNA 遗传算法在进化过程中,由于交叉、变异等操作的随机性,有可能出现种群多样性下降的情况,从而会出现早熟收敛、陷入局部最优；在人工鱼群算法中加入混沌搜索,能进一步提高全局搜索能力。因此,本节研究了一种 DNA 遗传混沌人工鱼群算法优化 DNA 序列的频域加权多模盲均衡算法,进一步改善了频域加权多模盲均衡算法的性能。

1. 混沌人工鱼群算法

(1) 基本思想

AFSA 具有收敛速度快、寻优精度高等优点，但 AFSA 的种群初始化具有一定随机性，增加了陷入局部搜索的概率，而混沌序列具有遍历性和规律性等特点。若将人工鱼的初始位置向量映射到混沌变量中，通过迭代产生混沌序列，并作逆映射得到人工鱼群新的位置向量，并作为人工鱼的初始位置，这就是混沌人工鱼群算法。该算法利用混沌扰动摆脱局部搜索，提高了全局搜索能力。

(2) 算法步骤

步骤 1：人工鱼群种群初始化。

设人工鱼群初始种群 $\boldsymbol{f}_{\mathrm{AFS}}=(\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N)$，其中 $\boldsymbol{f}_i(i=1,2,\cdots,N)$ 为第 $i$ 条人工鱼的初始位置向量，$N$ 为人工鱼群中个体数量。

步骤 2：人工鱼群混沌初始化并进行逆映射得到新的位置向量。

通过 Logistic 映射，得人工鱼群混沌位置向量为

$$\boldsymbol{f}(k)=\frac{\boldsymbol{f}_i(k)-\boldsymbol{f}_{\min}}{\boldsymbol{f}_{\max}-\boldsymbol{f}_{\min}} \tag{8.1.16}$$

式中，$\boldsymbol{f}_{\min}<\boldsymbol{f}_i(k)<\boldsymbol{f}_{\max}$。

混沌位置第 $k$ 次与第 $k+1$ 次的迭代关系为

$$f_i(k+1)=\rho f_i(k)[1-f_i(k)] \tag{8.1.17}$$

式中，$\rho\in(2,4]$ 为 Logistic 参数。

对当前的混沌变量进行逆映射，得人工鱼群的新位置向量，并作为人工鱼群的初始位置。

步骤 3：计算适应度函数并进行行为选择。

计算每条人工鱼的适应度函数，将适应度最大值及其对应的位置向量分别记录在公告牌中。人工鱼群进行行为操作，人工鱼的当前位置向量发生改变。

步骤 4：计算适应度方差 $\sigma^2$。

如果 $\sigma^2<\varepsilon$，说明算法陷入局部搜索，转到步骤 5 进行混沌扰动；否则转步骤 6。适应度方差定义为

$$\sigma^2=\sum_{i=1}^{D}\left(\frac{\mathrm{Fit}(\boldsymbol{f}_i)-\mathrm{Fit}_{\mathrm{avg}}}{\mathrm{Fit}}\right)^2 \tag{8.1.18}$$

式中，$D$ 为人工鱼位置向量的维数；$\mathrm{Fit}(\boldsymbol{f}_i)$ 为第 $i$ 条人工鱼的适应度函数值；$\mathrm{Fit}_{\mathrm{avg}}$ 为当前人工鱼群的适应度平均值；Fit 为归一化因子，其计算公式为

$$\mathrm{Fit}=\begin{cases}\max\limits_{1\leqslant i\leqslant D}|\mathrm{Fit}_i-\mathrm{Fit}_{\mathrm{avg}}|, & \max|\mathrm{Fit}_i-\mathrm{Fit}_{\mathrm{avg}}|>1\\ 1, & \text{其他}\end{cases} \tag{8.1.19}$$

步骤 5：混沌扰动。

在视野范围内，对人工鱼的位置向量进行混沌搜索，则新的位置向量为

$$\boldsymbol{f}_{i|\mathrm{next}}=\boldsymbol{f}_i+\Delta_i\boldsymbol{e}_i \tag{8.1.20}$$

$$\Delta_i=-\mathrm{Visual}+\rho\cdot\mathrm{Visual}\cdot q_d \tag{8.1.21}$$

式中，$q_d$ 为经过混沌迭代后的混沌变量；$\boldsymbol{e}_i$ 为单位向量。

步骤 6：更新公告牌。

计算每条人工鱼新的位置向量对应的适应度值，将适应度最大值与公告牌中保存的适应度最大值进行比较；如果当前适应度值大，则用当前适应度最大值及其对应的人工鱼位置向量更新公告牌中内容。

步骤 7：终止条件判断。

判断迭代次数是否达到设定值，若满足条件，则输出公告牌中的位置向量；若没有达到条件，跳转到步骤 3。

### 2. DNA 混沌人工鱼群算法

(1) 基本原理

DNA 遗传算法在进化过程中，由于交叉、变异等操作的随机性，有可能出现种群多样性下降的情况，从而会出现早熟收敛，陷入局部最优。为了提高 DNA 遗传算法的搜索性能，将交叉变异 DNA 遗传算法和混沌人工鱼群

算法(chaos artificial fish swarm algorithm,CAFSA)相结合,研究了一种交叉变异 DNA 遗传混沌人工鱼群算法(crossover and mutation DNA genetic algorithm chaos artificial fish swarm algorithm,DNA-GA-CAFSA),弥补了人工鱼群寻优精度不高、计算复杂度高、易于陷入局部搜索等不足。

(2) 基本步骤

步骤 1:人工鱼群种群初始化。

设人工鱼群初始种群 $\boldsymbol{f}_{\mathrm{AFS}}=(\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N)$,其中 $\boldsymbol{f}_i(i=1,2,\cdots,N)$为第 $i$ 条人工鱼的初始位置向量,$N$ 为人工鱼群中个体数量。

步骤 2:混沌人工鱼群算法。

通过混沌人工鱼群算法,得到新种群的位置向量。

步骤 3:反解码操作。

将混沌人工鱼群新的位置向量按式(7.1.2)和式(7.1.3)进行反解码,将十进制位置向量转换成四进制 DNA 编码序列,产生 DNA 编码序列种群。

步骤 4:对种群分组。

计算 DNA 编码序列种群的适应度值,并按照适应度值大小分为优质种群和劣质种群。

步骤 5:交叉操作。

从优质种群和劣质种群中各选择出两个 DNA 编码序列作为父体进行置换交叉操作,产生新的 DNA 编码序列,并代替父体,再各选择出一个 DNA 编码序列作为父体进行转位交叉操作,产生新的 DNA 编码序列,并代替父体,形成新的 DNA 编码序列种群。

步骤 6:变异操作。

在种群中随机选择出一个 DNA 编码序列作为父体进行变异操作,产生新的 DNA 编码序列,并代替父体,形成新的种群。

步骤 7:更新公告牌。

将 DNA 编码序列种群进行解码,得到十进制位置向量,即人工鱼群新的位置向量,计算每条混沌人工鱼新的位置向量对应的适应度值,将当前适

应度最大值与公告牌中保存的适应度最大值进行比较；如果当前适应度值大，则用当前适应度最大值及其对应的混沌人工鱼位置向量更新公告牌中内容。

步骤 8：终止条件判断。

判断 DNA 遗传混沌人工鱼群算法的迭代次数是否达到最大迭代次数，若满足条件，则输出公告牌中的位置向量，终止算法；若没有达到条件，跳转到步骤 2。

因此，上述步骤 1～步骤 8 构成 DNA 遗传混沌人工鱼群算法。其中，人工鱼群算法解决了宏观全局寻优问题；而混沌算法处理了微观局部加速问题；DNA 遗传算法利用置换交叉操作、转位操作、重构交叉操作等手段对全局最优解进行有效搜索。将三者融合使用，可以充分发挥各自的优势，获得更好的优化效果。

### 3. 基于 DNA 混沌人工鱼群算法优化 DNA 序列的频域加权多模盲均衡算法

(1) 频域加权多模盲均衡算法

频域加权多模算法（frequency domain weighted multi-modulus algorithm，FWMMA）是时域多模盲均衡算法在频域中的拓展。此时，误差函数的实部和虚部可定义为

$$\begin{cases} E_{\mathrm{R,FWMMA}}(k) = (Z^2_{\mathrm{R,FWMMA}}(k) - R^2_{\mathrm{R,FFT}}) \\ E_{\mathrm{I,FWMMA}}(k) = (Z^2_{\mathrm{I,FWMMA}}(k) - R^2_{\mathrm{I,FFT}}) \end{cases} \tag{8.1.22}$$

式中，$Z_{\mathrm{R,FWMMA}}(k)$和 $Z_{\mathrm{I,FWMMA}}(k)$分别表示频域均衡器输出信号的实部和虚部；$R_{\mathrm{R,FFT}}$和 $R_{\mathrm{I,FFT}}$表示发射信号统计模值实部和虚部的快速傅里叶变换，可表示为

$$\begin{cases} R^2_{\mathrm{R,FFT}} = \mathrm{FFT}\{E[|a_{\mathrm{R}}(k)|^4]/E[|a_{\mathrm{R}}(k)|^{2+\lambda}]\} \\ R^2_{\mathrm{I,FFT}} = \mathrm{FFT}\{E[|a_{\mathrm{I}}(k)|^4]/E[|a_{\mathrm{I}}(k)|^{2+\lambda}]\} \end{cases} \tag{8.1.23}$$

FWMMA 的代价函数为

$$J_{\mathrm{FWMMA}} = E\{[E_{\mathrm{R,FWMMA}}(k)]^2 + [E_{\mathrm{I,FWMMA}}(k)]^2\} \tag{8.1.24}$$

因此,FWMMA 的权向量更新公式为

$$\begin{cases} \boldsymbol{F}_{\mathrm{R,FWMMA}}(k+1) = \boldsymbol{F}_{\mathrm{R,FWMMA}}(k) - \mu Z_{\mathrm{R,FWMMA}}(k) E_{\mathrm{FWMMA,R}}(k) \boldsymbol{Y}^*_{\mathrm{R,FWMMA}}(k) \\ \boldsymbol{F}_{\mathrm{I,FWMMA}}(k+1) = \boldsymbol{F}_{\mathrm{I,FWMMA}}(k) - \mu Z_{\mathrm{I,FWMMA}}(k) E_{\mathrm{FWMMA,I}}(k) \boldsymbol{Y}^*_{\mathrm{I,FWMMA}}(k) \end{cases} \tag{8.1.25}$$

式中,$\boldsymbol{F}_{\mathrm{R,FWMMA}}(k)$和 $\boldsymbol{F}_{\mathrm{I,FWMMA}}(k)$分别表示频域权向量的实部和虚部;$\boldsymbol{Y}^*_{\mathrm{R,FWMMA}}(k)$和 $\boldsymbol{Y}^*_{\mathrm{I,FWMMA}}(k)$为频域均衡器输入信号实部和虚部的共轭。

FWMMA 通过傅里叶变换将时域盲均衡算法转换成频域算法,减少了计算量。

(2) DNA 遗传混沌人工鱼群优化 DNA 序列的频域加权多模盲均衡算法

DNA 遗传算法不仅具有传统遗传算法的固有优点,而且编码方式灵活、编码精度较高。但是,目前 DNA-GA 的交叉算子是模拟 DNA 分子的重组操作设计的,太过单一。AFSA 种群的初始化有一定随机性,增加了陷入局部搜索的概率,易于陷入局部搜索中。因此,将 DNA-CAFSA 和 FWMMA 相结合,研究了一种 DNA 混沌人工鱼群算法优化 DNA 序列的频域多模盲均衡算法(DNA chaos genetic artificial fish swarm algorithm based frequency domain weighted multi-Modulus blind equalization algorithm,DNA-CAFSA-FWMMA),其原理如图 8.7 所示。

算法基本步骤如下。

步骤 1:种群初始化并解码。

设人工鱼群初始种群 $\boldsymbol{f}_{\mathrm{AFS}}=(\boldsymbol{f}_1,\boldsymbol{f}_2,\cdots,\boldsymbol{f}_N)$,其中 $\boldsymbol{f}_i(i=1,2,\cdots,N)$为第 $i$ 条人工鱼的初始位置向量,$N$ 为人工鱼群中个体数量。再随机设置一组 DNA 编码序列 $s_0$ 作为计算相似度的对比序列。将 DNA 编码序列按式(7.1.2)和式(7.1.3)进行解码,得到十进制位置向量种群。

步骤 2:确定两个适应度函数。

多模盲均衡算法的目的是得到均衡器最优权向量,即代价函数处于最

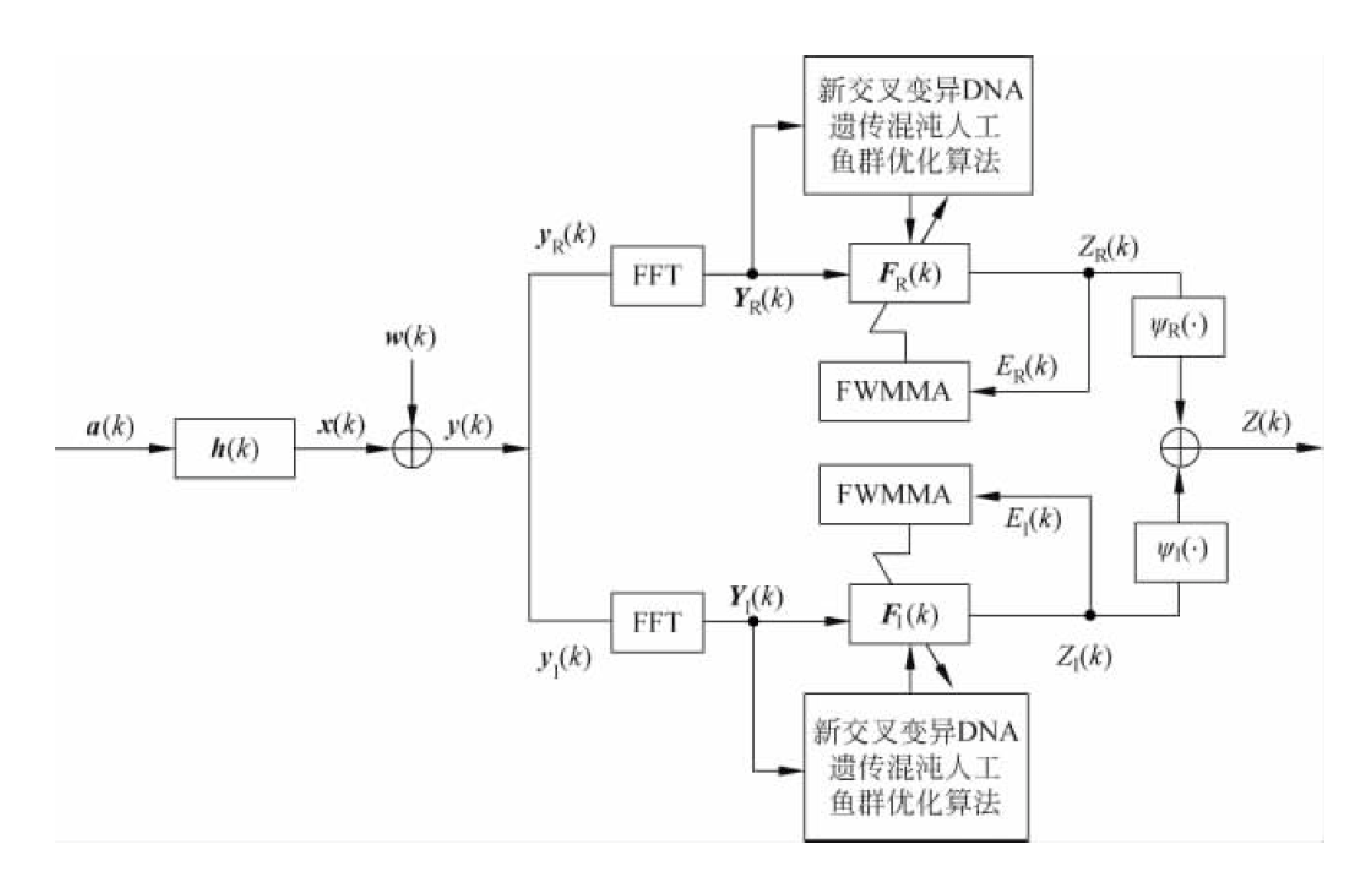



图 8.7　DNA-CAFSA-FWMMA 原理

小值状态，而人工鱼群算法的目的是寻找食物浓度最大值所对应的人工鱼个体的位置向量。

因此，第一个适应度函数为盲均衡器代价函数的倒数，即

$$Y_1(\boldsymbol{f}_i) = \frac{1}{J(\boldsymbol{f}_i)},\quad i = 1,2,\cdots,N \tag{8.1.26}$$

式中，$J(\boldsymbol{f}_i)=J_{\mathrm{FWMMA}}(\boldsymbol{f}_i)$为盲均衡器的代价函数；$\boldsymbol{f}_i$ 为新交叉变异 DNA 遗传人工鱼群算法优化的第 $i$ 条人工鱼的位置向量。

第二个适应度函数是采用加权平均值法来处理汉明距离约束项的函数，即

$$Y_2(\boldsymbol{f}_i) = w_1 \cdot H_1(\boldsymbol{f}_i,\boldsymbol{f}_i) \tag{8.1.27}$$

式中，$\boldsymbol{f}_i$ 对应于 DNA 编码序列中的第 $i$ 个 DNA 序列。为计算方便，设 $w=1$。

步骤 3：计算两个适应度函数。

按式(8.1.26)计算 DNA 遗传混沌人工鱼群算法的位置向量对应的第一个适应度值，并将第一个适应度最大值及其对应的十进制位置向量记录

在公告牌 1 中；再按式(8.1.27)计算 DNA 编码序列位置向量对应的第二个适应度值，并将第二个适应度最大值及其对应的 DNA 编码序列的位置向量记录在公告牌 2 中。

步骤 4：执行 DNA 遗传混沌人工鱼群算法并更新公告牌 2。

将第二个适应度函数式(8.1.27)作为 DNA 遗传混沌人工鱼群算法的适应度函数，对十进制位置向量进行混沌人工鱼群的觅食、聚群、追尾等行为，将十进制位置向量进行反解码，得到四进制 DNA 编码序列，进行 DNA 遗传算法的交叉、变异操作，并计算其第二个适应度值，与公告牌 2 中保存的原第二个适应度最大值进行比较，如果第二个适应度值大，则用第二个适应度值及其对应的四进制 DNA 编码序列更新公告牌 2 原先保存的内容。

步骤 5：更新公告牌 1。

在一次迭代之后，将公告牌 2 中的 DNA 序列进行解码，再利用第一个适应度函数式(8.1.26)，计算其第一个适应度值，与公告牌 1 中保存的原第一个适应度最大值进行比较，如果第一个适应度值大，则用第一个适应度值及其对应的十进制位置向量更新公告牌 1 原先保存的内容。

步骤 6：判断是否达到最大迭代次数。

如果当前迭代次数已经达到设定值，则将公告牌 1 中十进制位置向量作为最优位置向量输出，并将其作为 DNA-CAFSA-FWMMA 初始最优权向量的实部和虚部；反之，则返回步骤 4。

### 8.1.4 仿真实验

为了检验 DNA-CAFSA-FWMMA 的性能，将 WMMA 和 FWMMA 作为对比对象，进行仿真实验。

WMMA 和 FWMMA 的均衡器权长为 11，信噪比 SNR＝20dB，样本个数 $N$＝10000。

DNA-CAFSA-FWMMA 的种群规模取 30，人工鱼群的拥挤度因子取 0.68，人工鱼步长取 0.2，视野取 0.8，人工鱼群的迭代次数为 100，信道噪声为

高斯白噪声，混合相位水声信道 $\boldsymbol{h}=[0.3132 \quad -0.1040 \quad 0.8908 \quad 0.3134]$。

**【实验 8.3】** 发射信号为 64QAM 时，步长 $\mu_{\text{WMMA}}=0.00004$、$\mu_{\text{FWMMA}}=0.00003$，$\mu_{\text{DNA-CAFS-FWMMA}}=0.00003$；发射信号为 128QAM 时，$\mu_{\text{WMMA}}=0.65\times10^{-6}$、$\mu_{\text{FWMMA}}=0.54\times10^{-6}$、$\mu_{\text{DNA-CAFS-FWMMA}}=0.4\times10^{-6}$。在保证眼图完全清晰睁开的前提下，蒙特卡罗仿真结果如图 8.8 所示。

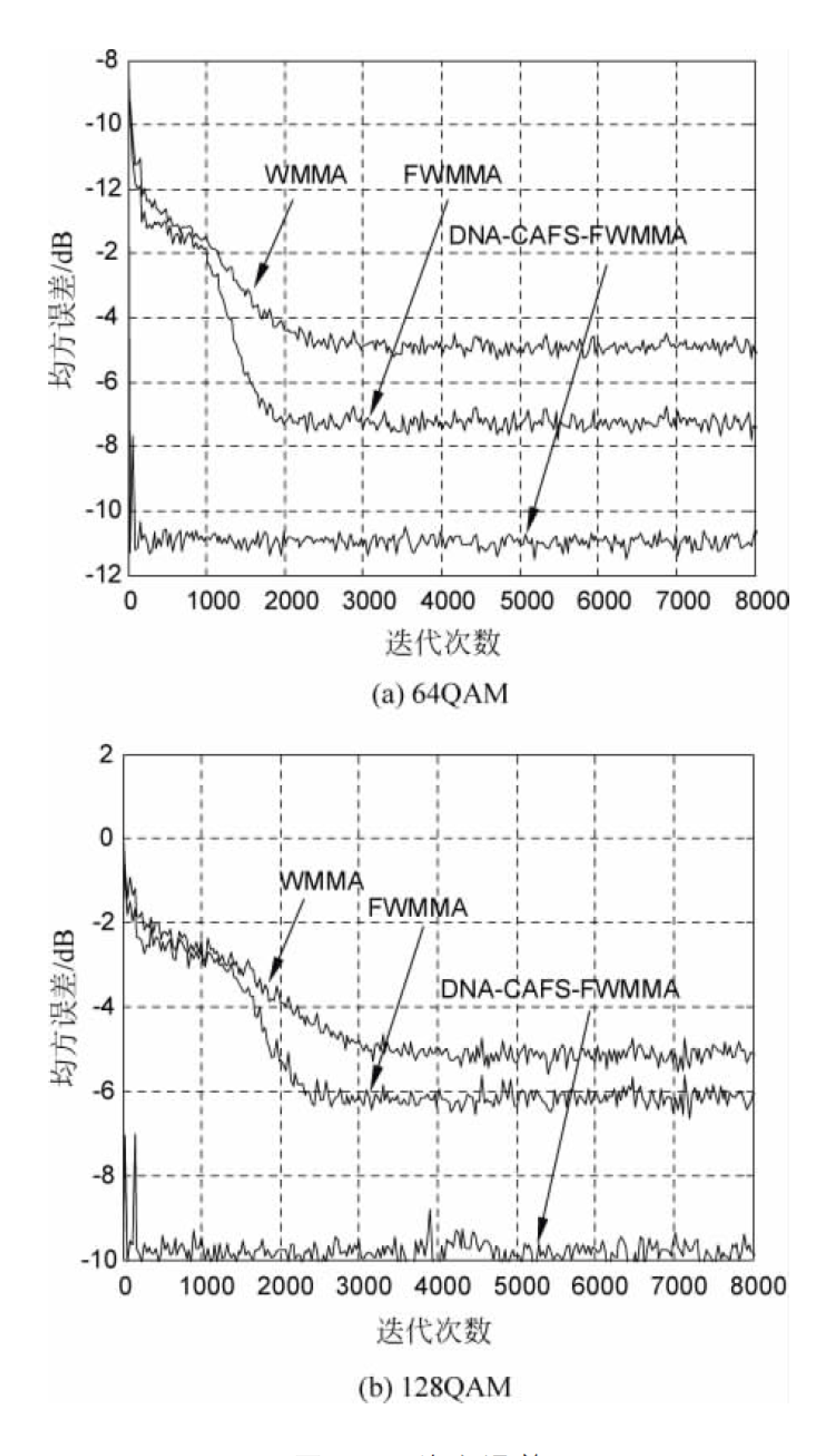



图 8.8 均方误差

图 8.8(a)表明,在收敛速度上,DNA-CAFSA-FWMMA 比 FWMMA 快了大约 1800 步,比 WMMA 快大约 2800 步;从均方误差上看,DNA-CAFSA-FWMMA 比 WMMA 减少了 4dB,比 FWMMA 减少了 2.2dB。图 8.8(b)表明,从收敛速度上看,DNA-CAFSA-FWMMA 比 WMMA 快大约 2600 步,比 FWMMA 快大约 2000 步;从均方误差上看,DNA-CAFSA-FWMMA 比 WMMA 减少了 3dB,比 FWMMA 减少了 1.7dB。

## 8.2 基于 DNA 遗传蛙跳算法优化的盲均衡算法

### 8.2.1 混合蛙跳算法

#### 1. 混合蛙跳算法的基本思想

混合蛙跳算法(shuffled frog leaping algorithm, SFLA)[36-45]是由 Eusuff 和 Lansey 在 2003 年首次提出,它是一种基于群体行为的粒子群优化算法(particle swarm optimize, PSO),通过模拟青蛙种群觅食的过程而建立。动物在生存觅食时一般会遵守对准规则、内聚规则和分割规则。对准规则是指群体中的个体都会向一个方向靠拢,内聚规则是指个体会向附近的中心靠拢,这两个规则是个体之间的信息交流;分割规则是指个体运动时和别的个体之间不会太拥挤,分割规则是个体自己在总结经验。

混合蛙跳算法是通过模拟青蛙种群觅食行为而得到的算法。在一片池塘中有很多只青蛙,每个青蛙都有一个位置向量,通过每个位置向量可得到相应青蛙的适应度值,即青蛙的解;将池塘中的青蛙可看作许多个子种群,每个子种群包含许多只青蛙且这些青蛙在觅食过程中可以互相交换信息,在子种群中,距离食物最远的青蛙可以和距离食物最近的青蛙进行交流从而向靠近食物方向移动;各个子种群间之间也可以交换信息,青蛙通过种群内和种群间信息交流向食物靠近,最终会有一个与食物距离最近的青蛙,这个青蛙个体就是最优解。

### 2. 混合蛙跳算法模型建立

随机生成 $N$ 个青蛙,每个青蛙维数为 $D$,由这些青蛙组成初始青蛙种群 $\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$,表示第 $i(1\leqslant i\leqslant N)$只青蛙的位置向量。对每一只青蛙计算出它的适应度值,然后将青蛙按照适应度值从大到小排列起来,再将青蛙种群分成 $M$ 个子种群。从适应度值最大的青蛙开始依次往后分到这 $M$ 个子种群中。例如,第一只分到第一个子种群中,第二只分到第二个子种群中,按照此规律,第 $M+1$ 只青蛙分到第一个子种群中,第 $M+2$ 只青蛙到第二子种群中,等等,最后将所有的青蛙都分到对应的子种群中。

子种群中的最差青蛙和最优青蛙分别表示为 $X_w$ 和 $X_b$,$X_g$ 是整个种群中最佳青蛙,对子种群中最差青蛙进行更新。青蛙的蛙跳步长为

$$\Omega_i = \text{rand}()\cdot(X_b - X_w),\quad (\|\Omega_{\min}\| \leqslant \|\Omega_i\| \leqslant \|\Omega_{\max}\|) \tag{8.2.1}$$

青蛙的位置更新公式为

$$X_{w,\text{new}} = X_w + \Omega_i \tag{8.2.2}$$

式中,$\Omega_i$ 表示青蛙的更新步长;rand()为均匀分布在[0,1]之间的随机数;$\|\Omega_{\max}\|$ 表示在允许范围内的最大蛙跳步长;$\|\Omega_{\min}\|$ 表示在允许范围内的最小蛙跳步长。根据式(8.2.1)与式(8.2.2)更新得到最差青蛙的位置向量为 $X_{w,\text{new}}$,计算适应度值,然后与原来的 $X_w$ 比较,若优于 $X_w$,则用更新后的最差青蛙位置取代原来群体中的当前最差青蛙位置;否则,最差青蛙的蛙跳步长公式为

$$\Omega_{i'} = \text{rand}()\cdot(X_g - X_w)\quad (\|\Omega_{\min}\| \leqslant \|\Omega_{i'}\| \leqslant \|\Omega_{\max}\|) \tag{8.2.3}$$

最差青蛙的位置更新公式为

$$X_{w,\text{new}} = X_w + \Omega_{i'} \tag{8.2.4}$$

执行更新策略式(8.2.3)与式(8.2.4)。若 $X_{w,\text{new}}$的适应度值比 $X_w$ 的小,就随机生成一个新的最差青蛙个体代替 $X_w$。

对种群中所有的子种群执行上述更新过程，即完成局部寻优后，计算种群中所有青蛙的适应度值，再将所有青蛙按照适应度值从大到小排列起来，然后重新划分子种群，每个子种群再执行局部搜索，循环执行上述过程直到达到设置的迭代次数才完成整体的寻优过程。

3. 混合蛙跳算法操作

混合蛙跳算法具有思想简单、寻优能力强、实验参数少、计算速度快等特点，但同时有易陷入局部最优的缺点。

混合蛙跳算法的具体步骤如下：

步骤 1：青蛙种群初始化及各参数设置。

随机生成 $N$ 只青蛙，每个青蛙的维数为 $D$，这些青蛙组成初始青蛙种群，第 $i(1\leqslant i\leqslant N)$只青蛙的位置向量可以表示为 $\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$。

步骤 2：计算所有青蛙的适应度值。

由适应度函数计算青蛙种群中所有青蛙的适应度值，并按适应度值从大到小排列起来，从而得到排在第一位的全局最优青蛙个体。

步骤 3：种群分组。

根据蛙跳算法模型中分组规则对全部青蛙进行分组，即分成 $M$ 组，每组包含 $K$ 只青蛙的子种群。

步骤 4：子种群内寻优。

每个子种群执行 $K$ 次寻优。首先确定子种群中最优青蛙个体和最差青蛙个体，利用步长更新公式对最差青蛙个体进行位置更新，如果通过这种更新方法不能产生更好的青蛙，用全局最优青蛙个体代替这个最差青蛙位置，如果还是达不到要求，就随机产生一个青蛙个体来替代原来的最差青蛙个体。

步骤 5：混合所有青蛙。

完成步骤 4 后，根据适应度函数计算出所有青蛙的适应度值，然后从大到小排列起来，可以得到排在第一位的最优青蛙个体，并用这个最优青蛙个体取代原来的最优青蛙个体。

步骤 6：终止判断。

若总迭代次数达到设置的最大迭代次数，则终止程序，输出最优青蛙个体，否则重复执行步骤 3～步骤 5，直到满足最大迭代次数。

算法流程，如图 8.9 所示。

在混合蛙跳算法中，参数设置会影响最后的寻优结果。例如，种群的大小 $N$，它是影响寻优效果最明显的参数，$N$ 的设置与算法寻找全局最优解的几率相关，$N$ 越大，找到或接近全局最优的概率也就越大。再例如，子种群大小会影响局部搜索的性能，若这个参数设置过小，混合蛙跳算法局部寻优能力强的特点就不能发挥出来，使寻优结果不精确。另外，子种群进化代数的选择会影响青蛙间的信息交流，若这个参数设置过小，会使子种群中青蛙的信息交流变少，如果设置过大，则算法易于陷入局部最优。这些参数的设置，目前大部分都是仿真实验得出来的。

### 8.2.2　DNA 遗传蛙跳算法

由于混合蛙跳算法易于陷入局部最优，且 DNA 遗传算法全局搜索能力强，为了提高混合蛙跳算法的性能，采用 DNA 遗传算法优化混合蛙跳算法，得到 DNA 遗传蛙跳算法（DNA genetic shuffled frog leaping algorithm，DNA-SFLA）。具体操作步骤[36]如下：

步骤 1：初始化青蛙种群，确定青蛙总数 $N$，进化代数 $G_{max}$。

步骤 2：计算青蛙个体适应度值，并按适应度值从大到小排列起来，排在第一位的作为最优个体，然后将排序后的青蛙种群前一半为优质种群（SuG）和后一半为劣质种群（InG），并将执行交叉操作产生的青蛙个数设为零，即 $N_{cnew}=0$，对每只青蛙个体的十进制位置向量进行 DNA 编码。

步骤 3：执行普通交叉操作。从优质种群中随机选出两只青蛙作为父体，然后随机产生一个 0 到 1 的随机数，若此随机数小于交叉操作执行概率 $p_c$，则进行交叉操作，生成两个新的青蛙个体，即 $N_{cnew}=N_{cnew}+2$，直到新生

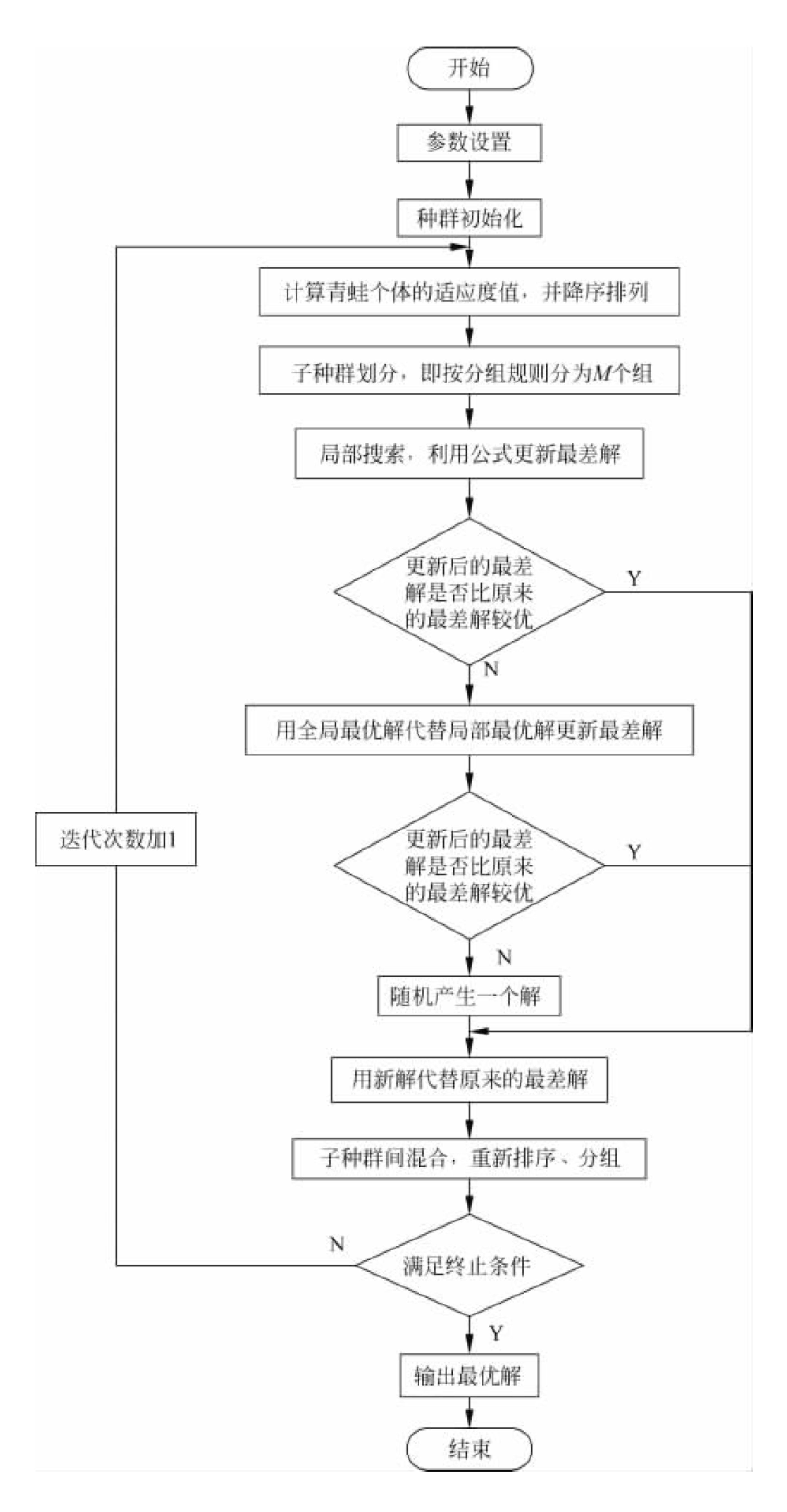



图 8.9　混合蛙跳算法流程

成的青蛙个体数 $N_{cnew}$ 大于 0.5$N$，则执行步骤 4，否则继续执行交叉操作。

步骤 4：执行变异操作。将步骤 3 产生的新青蛙个体全部放入青蛙种群中，然后对种群中所有个体（原始青蛙总个体数加上新产生的青蛙个体数）执行普通变异操作。对种群中每一个青蛙个体都随机产生一组 0 到 1 的随机数，其中每一个随机数与个体中每一个编码位相对应。若某一个随机数小于普通变异执行概率 $p_m$，则该随机数对应的碱基按 7.1.1 节所述的过程行进变异操作，从而产生一个新个体并代替变异前的个体。

步骤 5：执行选择操作，选择下一代种群。在执行完步骤 4 之后，首先进行 DNA 解码，然后对变异后的种群执行联赛选择操作，重复执行 $N-1$ 次，选择出 $N-1$ 个下一代个体，同时将未经交叉和变异操作之前的最优个体直接保留到下一代种群中，当前进化代数加 1。

步骤 6：判断是否达到进化代数。若是，就输出最优青蛙个体；否则，再执行步骤 2～步骤 6。DNA 遗传蛙跳算法流程，如图 8.10 所示。

### 8.2.3 基于 DNA 遗传蛙跳算法优化的常模盲均衡算法

#### 1. 算法原理

为了进一步提高 CMA 性能，现将 DNA-SFLA 应用到 CMA 中，得到基于 DNA 遗传蛙跳算法优化的常模盲均衡算法（DNA genetic shuffled frog leaping algorithm based CMA，DNA-SFLA-CMA）。算法步骤如下。

步骤 1：青蛙种群初始化并计算适应度值。

初始的青蛙种群是由随机生成的 $N$ 只 $D$ 维的青蛙组成，$\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$表示第 $i(1\leqslant i\leqslant N)$只青蛙的十进制位置向量。将代价函数的倒数定义为青蛙的适应度函数，即 $\mathrm{Fit}(\boldsymbol{X}_i)=\dfrac{1}{J_{\mathrm{CMA}}(\boldsymbol{X}_i)}$，$1\leqslant i\leqslant N$，计算种群中所有青蛙的适应度值，并按照适应度值从大到小进行排序，排在第一位的青蛙个体作为最优个体。将前一半青蛙作为优质种群，后一半作为劣质种群，令 $N_{cnew}$ 为执行生成的新的青蛙个体数，并将其初值设为零，种群进化代

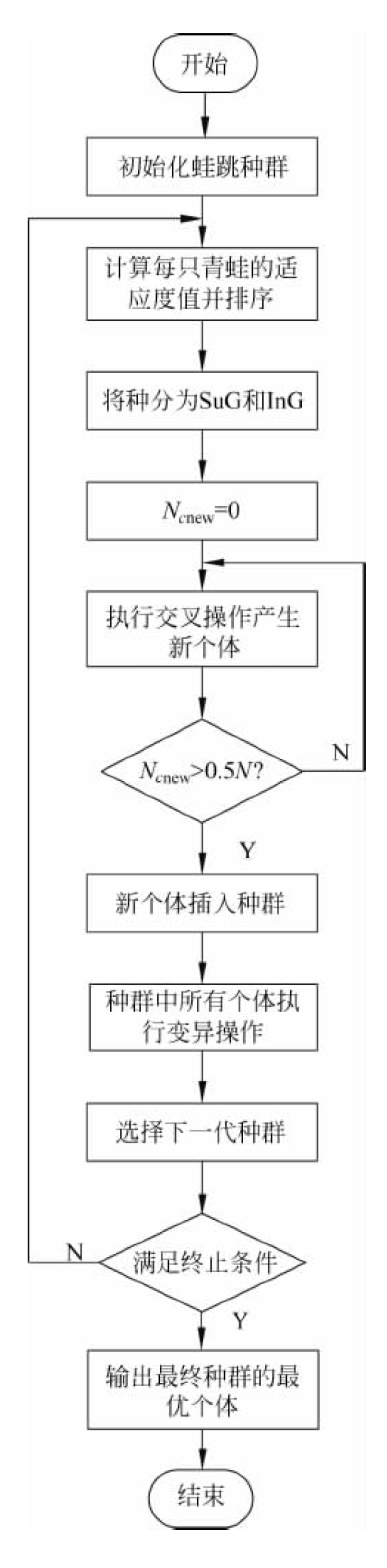



图 8.10　DNA 遗传蛙跳算法流程

数为 $G_{max}$。

步骤 2：青蛙种群 DNA 编码并进行交叉变异操作。

对每只青蛙个体的十进制位置向量进行 DNA 编码，然后对 DNA 编码后的个体进行交叉变异操作，交叉变异操作过程按 7.1.1 节所述方法进行。

步骤 3：青蛙种群 DNA 解码作并进行选择操作。

首先对青蛙种群进行 DNA 解码，然后对解码后的青蛙种群执行联赛选择操作，重复 $N-1$ 次，选择出 $N-1$ 个青蛙个体组成下一代种群，同时将未经交叉和变异操作之前的最优个体作为精英，也保留到下一代种群中，将当前进化代数加 1。

步骤 4：判断终止。

判断是否达到种群进化代数 $G_{max}$。若达到种群进化代数 $G_{max}$，则输出最优青蛙个体，并将最优青蛙个体位置向量赋值给 CMA 的初始权向量；否则，再执行步骤 2～步骤 3 的操作。

DNA-SFLA-CMA 原理及流程如图 8.11 所示。

## 2. 算法仿真

为了验证 DNA-SFLA-CMA 的性能，以 CMA 和 SFLA-CMA 作为对比对象，在信道 $\boldsymbol{h}=[0.005\quad 0.009\quad -0.024\quad 0.854\quad -0.218\quad 0.049\quad -0.016]$下进行仿真实验。

**【实验 8.4】** CMA 均衡器权长为 11，第 6 个抽头系数为 1，其余为 0，信道噪声为高斯白噪声，信噪比为 25dB，样本个数为 5000。SFLA-CMA 和 DNA-SFLA-CMA 的均衡器权长均为 11，青蛙总数为 55，子种群个数为 5，子种群进化 20 次，全局进化 20 次，DNA-SFLA-CMA 的交叉概率为 0.8，变异概率为 0.1，样本个数 $N=5000$。信源采用 16APSK 信号，步长 $\mu_{CMA}=0.000\,05$，$\mu_{SFLA\text{-}CMA}=0.000\,004$，$\mu_{DNA\text{-}SFLA\text{-}CMA}=0.000\,007$。500 次蒙特卡罗仿真结果如图 8.12 所示。

图 8.12 表明，DNA-SFLA-CMA 的星座图比 SFLA-CMA 和 CMA 的星座图更加集中、紧凑。在收敛速度上，CMA 在迭代 1500 次左右收敛，

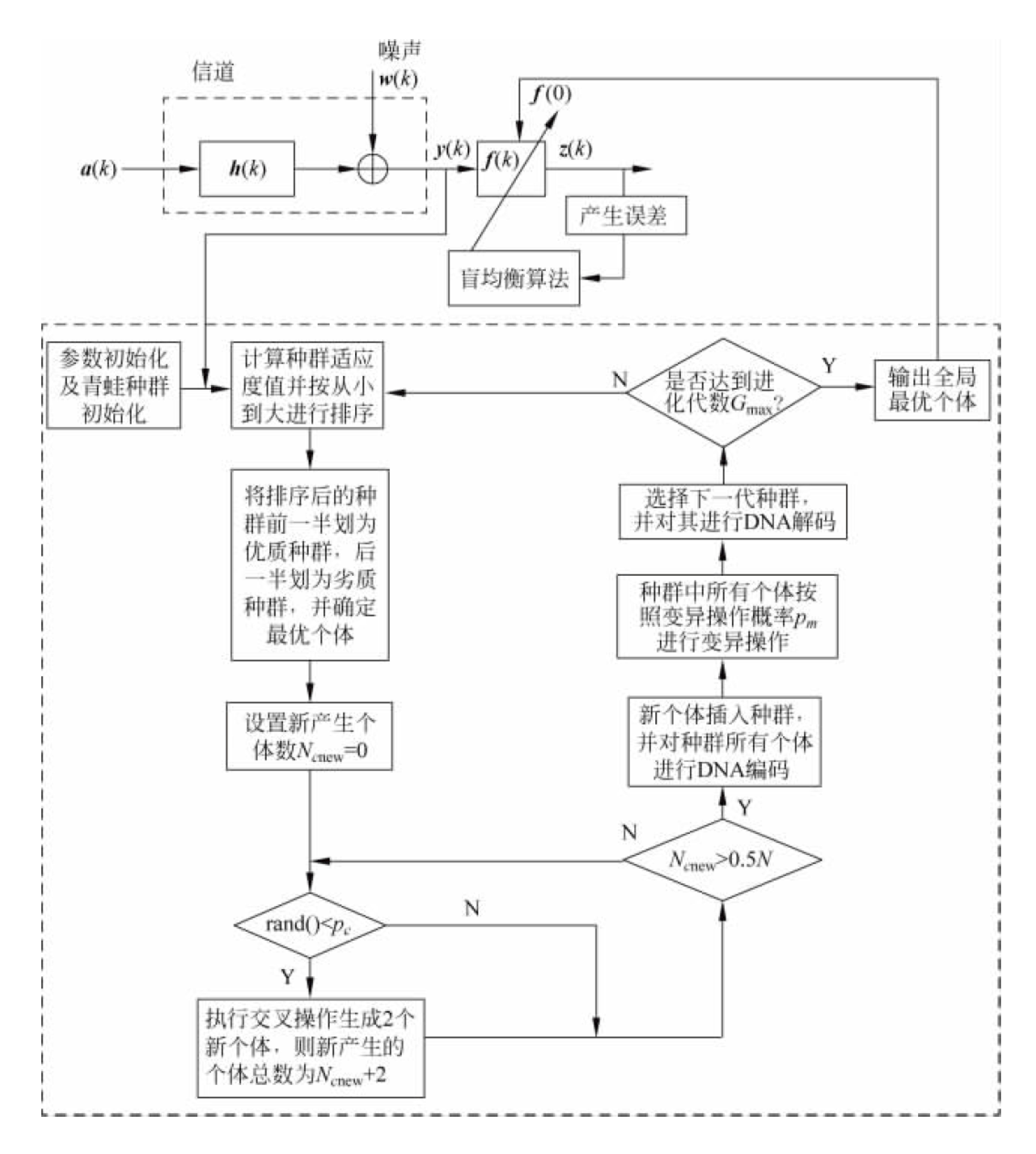
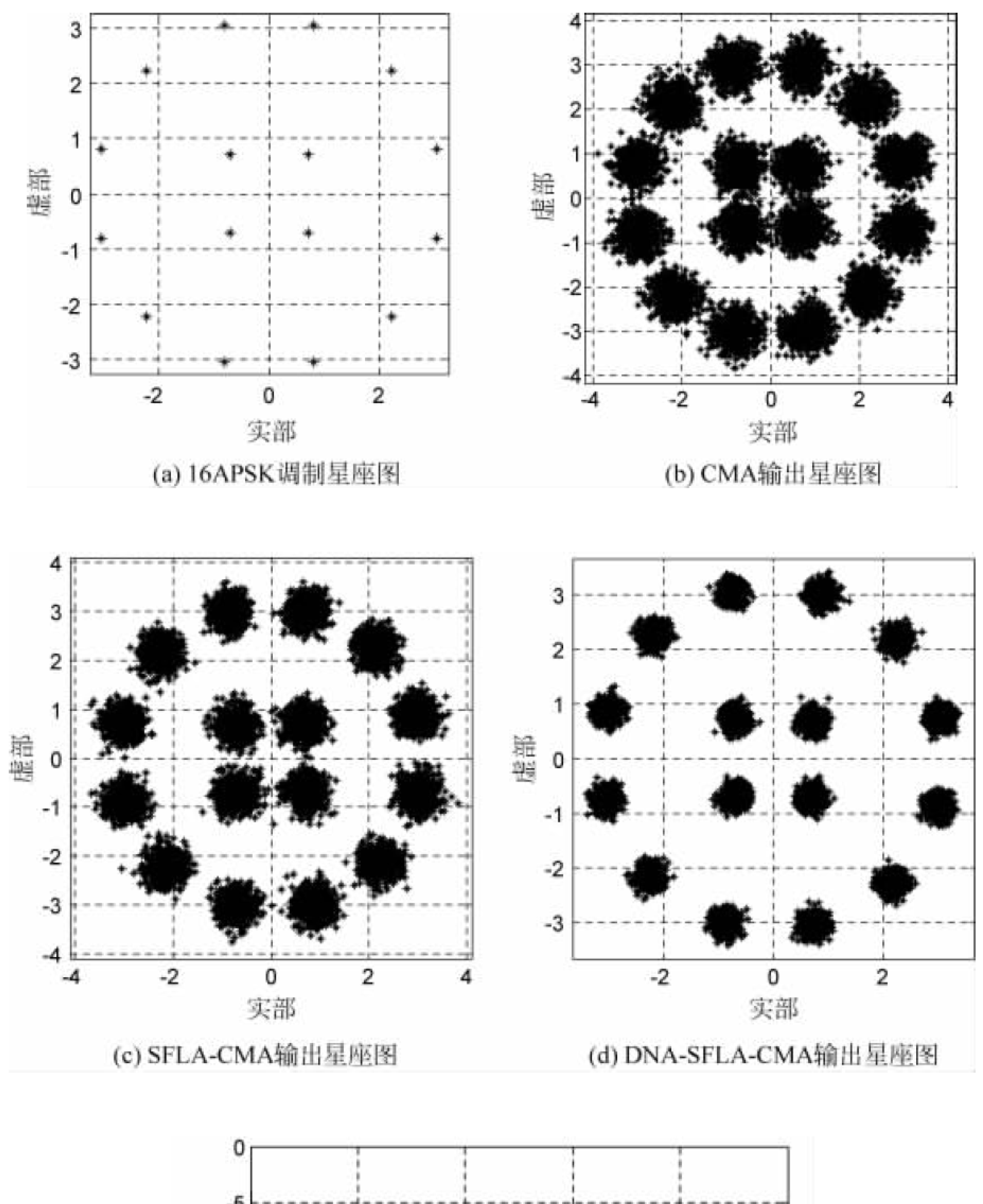



图 8.11　DNA-SFLA-CMA 原理及流程

SFLA-CMA 在迭代 500 次左右收敛，DNA-SFLA-CMA 在迭代 100 次左右收敛，DNA-SFLA-CMA 比 CMA 快了 1400 步左右，比 SFLA-CMA 快了 400 步左右；在稳态误差上，CMA 的均方误差稳定在－20dB 左右，SFLA-CMA 的均方误差稳定在－23dB 左右，DNA-SFLA-CMA 的均方误差稳定在－30dB 左右，DNA-SFLA-CMA 的均方误差比 CMA 的均方误差低了 10dB 左右，比 SFLA-CMA 的均方误差低了 7dB 左右。



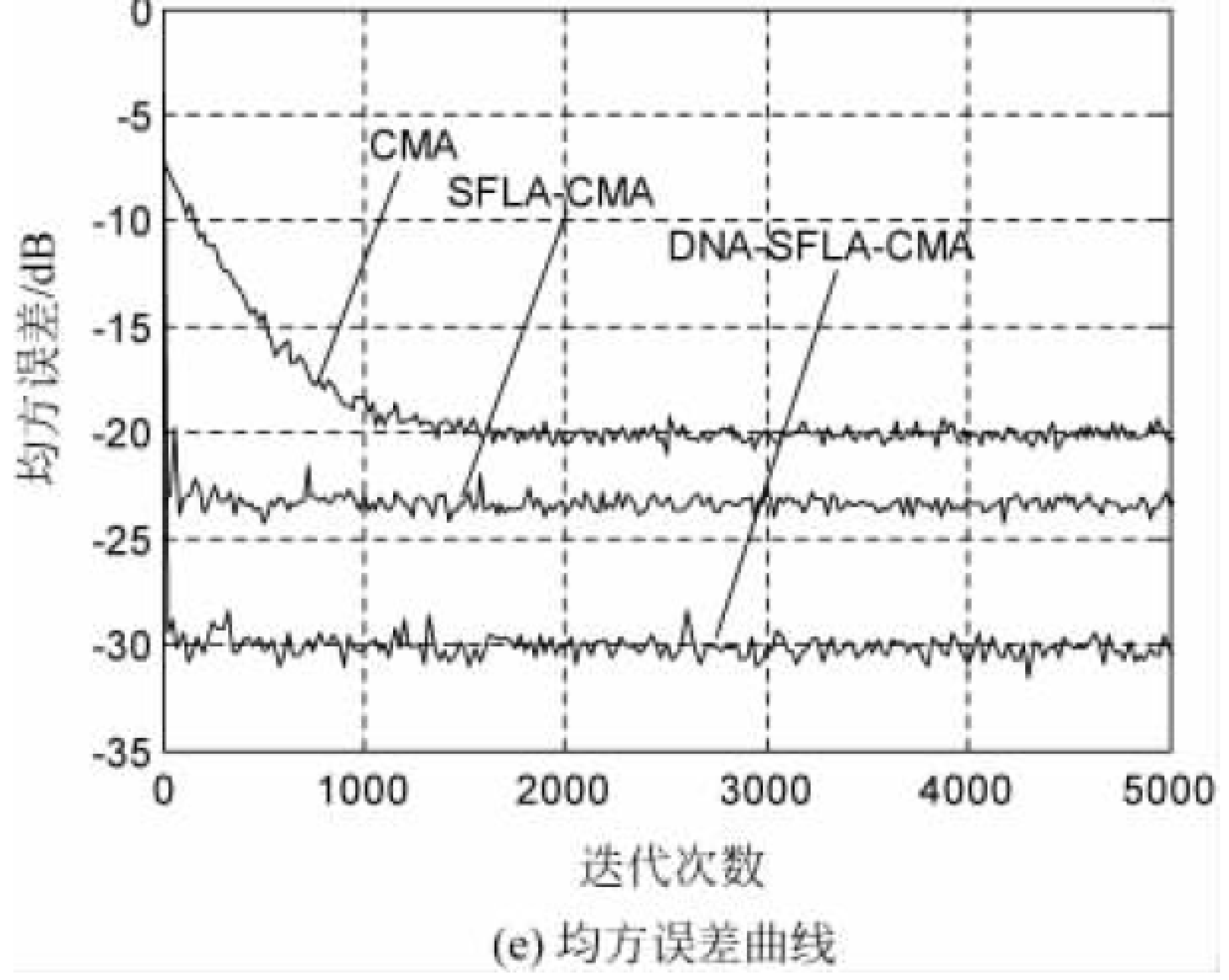




图 8.12 星座图均方误差

## 8.3 基于 DNA 遗传蝙蝠算法优化的分数间隔多模盲均衡算法

### 8.3.1 蝙蝠算法

蝙蝠算法(bat algorithm,BA)[46-51]是由 Yang Xin-she 于 2010 年提出的一种元启发式搜索算法。它来源于微型蝙蝠的回声定位功能。该算法在求解过程中,将蝙蝠个体所组成的群体从无序到有序进行演变,并调整响度、脉冲率、频率,搜索最优解。BA 不仅全局搜索能力强,而且收敛速度快。已经证明,在某些函数优化方面该算法优于遗传算法和标准粒子群算法。

1. 蝙蝠行为

蝙蝠发出的声呐可谓神奇,可以用来探测猎物,在黑暗中也能找到食物和栖息地,基于回声定位的声学原理,它们可以避免障碍和探寻猎物。研究表明,蝙蝠发出声波的频率为 25～150kHz,波长为 2～14mm,这样的波长和蝙蝠的猎物大小类似。而蝙蝠发出的声波响度是变化的。蝙蝠就是利用回声定位原理来探测物体的种类、距离和位置方向。

2. 蝙蝠算法

蝙蝠算法(BA)是一种基于种群的全局寻优算法,通过改变蝙蝠发出超声波的频率、频度、响度来搜索全局最优位置。在一个 $D$ 维空间中,蝙蝠群中第 $i$ 只蝙蝠的速度 $\boldsymbol{V}_i(k)$和位置向量 $\boldsymbol{X}_i(k)$的更新公式为

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{8.3.1}$$

$$\boldsymbol{V}_i(k) = \boldsymbol{V}_i(k-1) + [\boldsymbol{X}_i(k) - \boldsymbol{X}_{\text{now-opt}}]f_i \tag{8.3.2}$$

$$\boldsymbol{X}_i(k) = \boldsymbol{X}_i(k-1) + \boldsymbol{V}_i(k) \tag{8.3.3}$$

式中,$\beta$ 为[0,1]上的随机数;$f_i$ 为超声波频率,控制速度的更新;$\boldsymbol{X}_{\text{now-opt}}$ 是

当前全局最优位置向量。

局部搜索过程中，一旦从所有蝙蝠个体的当前最优位置向量中任意选出一个位置向量 $\boldsymbol{X}_{\text{old}}(k)$，那么蝙蝠群的新位置向量 $\boldsymbol{X}_{\text{new}}(k)$就在 $\boldsymbol{X}_{\text{old}}(k)$附近随机产生，即

$$\boldsymbol{X}_{\text{new}}(k) = \boldsymbol{X}_{\text{old}}(k) + \varepsilon A(k)\boldsymbol{e}_A \tag{8.3.4}$$

式中，$\varepsilon$ 为[−1,1]上的随机数；$A(k)$为 $k$ 时刻蝙蝠群的平均响度；$\boldsymbol{e}_A$ 为单位向量。用$\boldsymbol{X}_{\text{new}}(k)$代替当前位置向量 $\boldsymbol{X}_i(k)$，返回全局搜索继续搜索猎物，避免产生局部最优。

搜索猎物过程中，起初，为了搜索范围更广，超声波响度 $A$ 大而频度 $r$ 低；发现猎物后，为了更精确定位，响度 $A$ 减小而频度 $r$ 增大。所以，响度 $A$ 和频度 $r$ 的更新公式为

$$A_i(k+1) = \alpha A_i(k) \tag{8.3.5}$$

$$r_i(k+1) = r(0)[1-\exp(-\gamma k)] \tag{8.3.6}$$

式中，$\alpha$ 为响度衰减系数；$\gamma$ 为频度增加系数，是大于 0 的常数；$r(0)$为最大频度。初始化时，所有蝙蝠的响度和频度都不相同，通常最大频度 $r(0)\in[0,1]$，最大响度 $A(0)\in[1,2]$。响度 $A$ 和频度 $r$ 随当前全局最优位置 $\boldsymbol{X}_{\text{now-opt}}$的改变进行更新，表示蝙蝠群向最优位置移动。

蝙蝠算法(BA)的搜索过程分为全局寻优过程和局部寻优过程。全局寻优过程用式(8.3.1)～式(8.3.3)来搜索全局最优位置向量；式(8.3.4)～式(8.3.6)表示局部寻优过程，利用 BA 特有的回波定位特性来避免搜索过程陷入局部最优，这是其他智能算法所不具有的，这也使得 BA 具有其他智能算法无可比拟的优势。

3. 实现步骤

假设求函数 $J(\boldsymbol{X})$，$\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_N\}^{\text{T}}$ 的最小值，种群大小为 $N$，蝙蝠算法的基本步骤如下。

步骤 1：确定目标函数 $J(\boldsymbol{X})$，$\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_N\}^{\text{T}}$。

步骤 2：初始化种群数量 $N$，迭代次数 $G_{\max}$，蝙蝠响应 $A$，脉冲率 $r$。

步骤 3：计算个体初始值 Fit($\boldsymbol{X}$)＝$J$($\boldsymbol{X}$)及最有初始值 val_best。

```
While(t < G_max)
  For i = 1 to N
    利用式(8.3,1)～式(8.3.3)产生新一代解 X_new
    If rand > r_i
        用式(8.3.4)产生一个局部解 X_new
    End if
    If rand < A_i &&J (X_new)< Fit( i)
      X_i = X_new
      Fit(X_i) = J (X_new)
      通过式(8.3.5)更新响度 A_i 和脉冲率 r_i
    End if
  End for
  排列蝙蝠并找到当前最佳 X_now-opt
End while
处理结果并可视化实验数据。
```

### 8.3.2 DNA 遗传蝙蝠算法

类似于 GA，DNA-GA 以解的串集搜索最优解的方式使其拥有其他算法无可比拟的全局搜索能力，而 BA 的回波定位特性又能使搜索过程避免陷入局部最优。因此，利用 DNA-GA 对蝙蝠的位置向量 $\boldsymbol{X}_i(k)$进行编码、交叉、变异、解码等一系列操作优化 BA 的搜索过程，提出了 DNA 遗传蝙蝠算法(DNA-GBA)，可以达到以更快的速度搜索到全局最优位置向量的目的。

DNA-GBA 的适应度函数为蝙蝠个体位置向量的函数，定义为

$$\text{Fit}(\boldsymbol{X}_i(k)) = J_{\text{DNA-GBA}}(\boldsymbol{X}_i(k)) \tag{8.3.7}$$

DNA-GBA 的实现流程如下。

步骤 1：参数初始化。随机产生一个蝙蝠种群，蝙蝠个体数量为 $N$，频率范围为$[f_{\min}, f_{\max}]$，最大响度为 $A(0)$，最大频度为 $r(0)$，响度衰减系数为 $\alpha$，频度增加系数为 $\gamma$，置换交叉概率 $p_z$，移位交叉概率 $p_y$，变异概率 $p_b$，维数为 $D$，搜索精度为 tol，最大迭代次数为 $G_{\max}$，各蝙蝠个体的位置向量为 $\boldsymbol{X}_i$。

步骤 2：计算适应度函数值。根据式(8.3.7)计算各位置向量的适应度值并将适应度值从大到小排列，其中，前一半对应的蝙蝠个体组成优质种群，后一半对应的蝙蝠个体组成劣质种群。适应度最大值对应的位置向量作为当前全局最优位置向量 $\boldsymbol{X}_{\text{now-opt}}$。

步骤 3：调整频率 $f_i$，利用式(8.3.2)和式(8.3.3)对所有蝙蝠的速度和位置向量进行更新，得到蝙蝠群更新后的位置向量 $\boldsymbol{X}_i(k)$。

步骤 4：产生一个服从均匀分布的随机频度 rand1 与第 $i$ 只蝙蝠的频度 $r_i$ 进行比较，若 rand1$>r_i$，利用式(8.3.4)对处于当前最优位置的蝙蝠个体随机扰动产生一个新的位置向量，替代第 $i$ 只蝙蝠的当前位置向量并继续搜索猎物。

步骤 5：产生一个服从均匀分布的随机响度 rand2 与第 $i$ 只蝙蝠的响度 $A_i$ 进行比较，若 rand2$<A_i$ 且 $J_{\text{DNA-GBA}}(\boldsymbol{X}_i(k))>J_{\text{DNA-GBA}}(\boldsymbol{X}_{\text{now-opt}})$，则用第 $i$ 只蝙蝠的当前位置向量 $\boldsymbol{X}_i(k)$ 替代当前最优位置向量 $\boldsymbol{X}_{\text{now-opt}}$，并利用式(8.3.5)和式(8.3.6)对 $A_i$、$r_i$ 分别进行更新。

步骤 6：DNA 碱基编码。采用 DNA 碱基编码方式对各蝙蝠个体的位置向量进行编码，得到位置向量 DNA 序列。

步骤 7：置换交叉操作。产生一个随机数 rand3$\in(0,1)$，与置换交叉概率 $p_z$ 比较，若 rand3$<p_z$，则执行置换交叉操作。

步骤 8：转位交叉操作。产生一个随机数 rand4$\in(0,1)$，与转位交叉概率 $p_y$ 比较，若 rand4$<p_y$，则执行转位交叉操作。

步骤 9：变异操作。产生一组与蝙蝠个体位置向量 DNA 序列维数相同的(0,1)上的随机数，这组随机数中的元素与位置向量 DNA 序列中的元素一一对应，将所有随机数分别与变异概率 $p_b$ 比较，若随机数小于 $p_b$，则执行变异操作。

步骤 10：解码。将经交叉、变异后得到的所有蝙蝠个体的位置向量 DNA 序列解码，用解码得到的位置向量计算适应度值，从大到小排列并划分优质种群和劣质种群。

步骤 11：选取当前全局最佳位置向量 $\boldsymbol{X}_{\text{now-opt}}$。适应度最大值对应的位

置向量即为当前全局最优位置向量。

步骤 12：达到最大迭代次数或搜索精度，则输出全局最优位置向量 $\boldsymbol{X}_{\text{now-opt}}$，否则转至步骤 3 继续搜索。

### 8.3.3 基于DNA遗传蝙蝠算法优化的分数间隔多模盲均衡算法

#### 1. 分数间隔多模盲均衡算法

为了进一步加快收敛速度并减小稳态误差，本节利用 DNA 遗传算法(DNA genetic algorithm，DNA-GA)优化蝙蝠位置，得到 DNA 遗传蝙蝠算法(DNA genetic bat algorithm，DNA-GBA)；利用 DNA-GBA 对分数间隔多模盲均衡算法的权向量进行初始优化，得到一种基于 DNA 遗传蝙蝠算法优化的分数间隔多模盲均衡算法(DNA genetic bat algorithm based fractionally spaced multi-modulus algorithm，DNA-GBA-FS-MMA)，如图 8.13 所示。

图 8.13(b)中，第 $p$ 条支路($p=0,1,\cdots,P-1$)各物理量的关系如下。

信道冲激响应为

$$\boldsymbol{h}^{(p)}(k)=\boldsymbol{h}[(k+1)M-p-1] \tag{8.3.8}$$

式中，$\boldsymbol{h}$ 为整个系统的信道。

盲均衡器输入信号为

$$\boldsymbol{y}^{(p)}(k)=\boldsymbol{a}(k)\boldsymbol{h}^{(p)}(k)+\boldsymbol{w}^{(p)}(k) \tag{8.3.9}$$

将输入信号 $\boldsymbol{y}^{(p)}(k)$分为实部与虚部分别进行处理，得到均衡器输出信号的实部与虚部分别为

$$z_{\text{R}}^{(p)}(k)=\boldsymbol{f}_{\text{R}}^{(p)}(k)\boldsymbol{y}_{\text{R}}^{(p)}(k) \tag{8.3.10}$$

$$z_{\text{I}}^{(p)}(k)=\boldsymbol{f}_{\text{I}}^{(p)}(k)\boldsymbol{y}_{\text{I}}^{(p)}(k) \tag{8.3.11}$$

输出信号为

$$z^{(p)}(k)=z_{\text{R}}^{(p)}(k)+\text{j}z_{\text{I}}^{(p)}(k) \tag{8.3.12}$$

误差信号的实部与虚部分别为

$$e_{\text{R}}^{(p)}(k)=z_{\text{R}}^{(p)}(k)(|z_{\text{R}}^{(p)}(k)|^2-R_{\text{R}}^2) \tag{8.3.13}$$

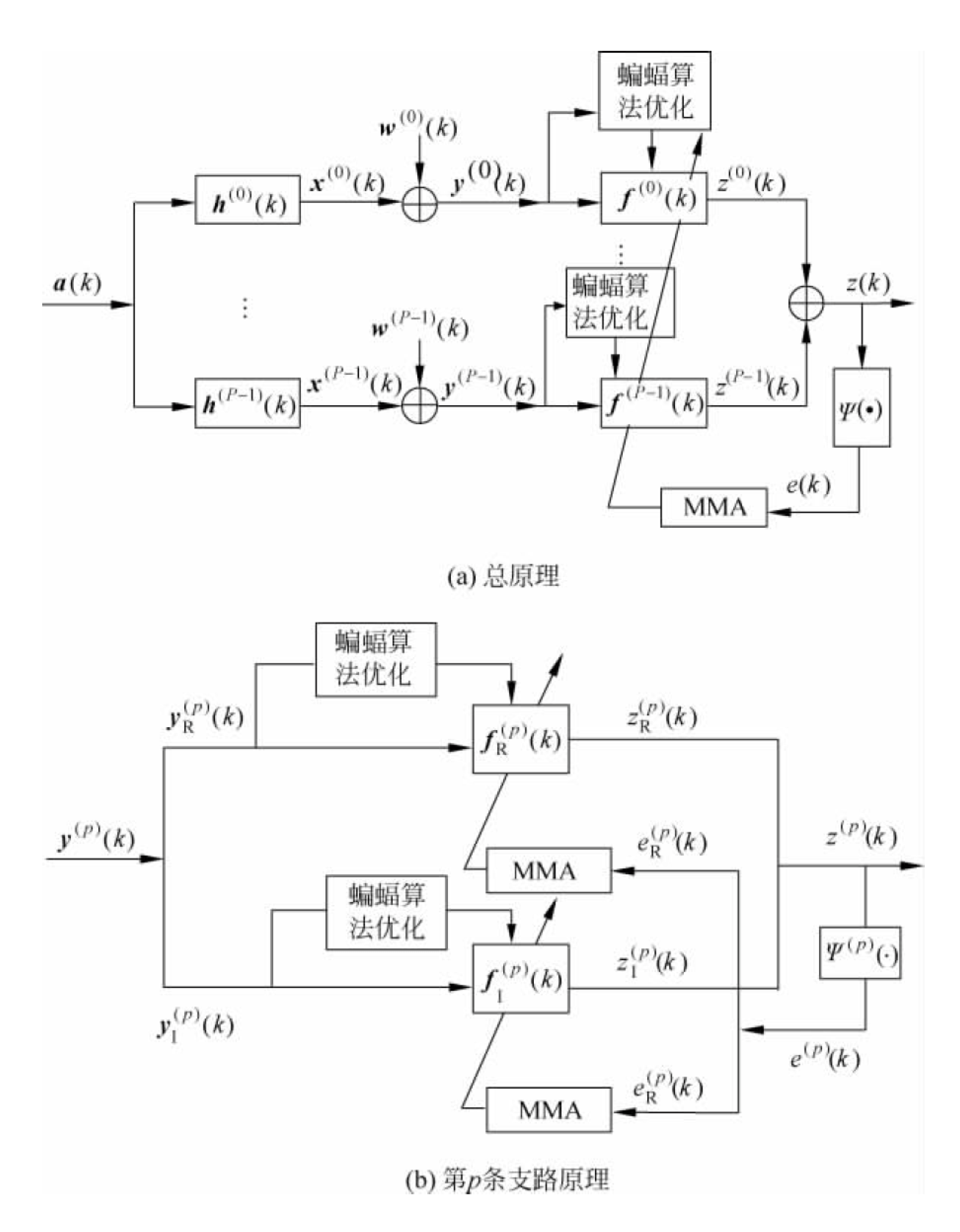



图 8.13　DNA-GBA-FS-MMA 原理

$$e_I^{(p)}(k) = z_I^{(p)}(k)(|z_I^{(p)}(k)|^2 - R_I^2) \tag{8.3.14}$$

式中，$R_R$ 和 $R_I$ 分别为发射信号 $a(k)$实部和虚部的统计模值。

图 8.13(b)中，第 $p$ 路 MMA 的代价函数定义为

$$\begin{aligned} J_{MMA}^{(p)}(k) &= J_R^{(p)}(k) + J_I^{(p)}(k) \\ &= E\{|z_R^{(p)}(k)|^2 - R_R^{(p)}]^2\} + \\ &\quad E\{[|z_I^{(p)}(k)|^2 - R_I^2]^2\} \end{aligned} \tag{8.3.15}$$

按照最速下降法，有

$$\begin{cases} \dfrac{\partial J_{\mathrm{R}}^{(p)}(k)}{\partial \boldsymbol{f}_{\mathrm{R}}^{(p)}(k)} = 4e_{\mathrm{R}}^{(p)}(k)\boldsymbol{y}_{\mathrm{R}}^{(p)}(k) \\ \dfrac{\partial J_{\mathrm{I}}^{(p)}(k)}{\partial \boldsymbol{f}_{\mathrm{I}}^{(p)}(k)} = 4e_{\mathrm{I}}^{(p)}(k)\boldsymbol{y}_{\mathrm{I}}^{(p)}(k) \end{cases} \tag{8.3.16}$$

因此，多模模块 $p$ 的权向量实部 $\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)$ 和虚部 $\boldsymbol{f}_{\mathrm{I}}^{(p)}(k)$ 迭代公式分别为

$$\boldsymbol{f}_{\mathrm{R}}^{(p)}(k+1) = \boldsymbol{f}_{\mathrm{R}}^{(p)}(k) - 4\mu e_{\mathrm{R}}^{(p)}(k)\boldsymbol{y}_{\mathrm{R}}^{(p)}(k) \tag{8.3.17}$$

$$\boldsymbol{f}_{\mathrm{I}}^{(p)}(k+1) = \boldsymbol{f}_{\mathrm{I}}^{(p)}(k) - 4\mu e_{\mathrm{I}}^{(p)}(k)\boldsymbol{y}_{\mathrm{I}}^{(p)}(k) \tag{8.3.18}$$

式中，$\mu$ 为步长。

分数间隔多模盲均衡算法（FS-MMA）的输出信号为

$$z(k) = z^{(0)}(k) + z^{(1)}(k) + \cdots + z^{(p)}(k) + \cdots + z^{(P-1)}(k) = \sum_{p=0}^{P-1} z_p(k) \tag{8.3.19}$$

2. DNA 遗传蝙蝠算法优化过程

DNA 遗传蝙蝠算法优化分数间隔多模盲均衡算法的实现流程如下。

步骤 1：初始化参数。除 DNA-GBA 须初始化外，还须初始化运行次数 runs，信噪比 SNR，均衡器抽头个数 $L_f$。

步骤 2：定义适应度函数。本节由 MMA 代价函数的倒数作 DNA-GBA 的适应度函数，即

$$J_{\mathrm{DNA\text{-}GBA}}(\boldsymbol{X}_i(k)) = \frac{1}{J_{\mathrm{MMA}}(k)} = \frac{1}{E\{[z_{\mathrm{R}}^2(k) - R_{\mathrm{R}}^2]^2\} + E\{[z_{\mathrm{I}}^2(k) - R_{\mathrm{I}}^2]^2\}} \tag{8.3.20}$$

步骤 3：利用 DNA-GBA 搜索全局最优位置向量 $\boldsymbol{X}_{\mathrm{opt}}$。按式（8.3.20）计算适应度函数，搜索过程为 DNA-GBA 实现流程。

步骤 4：把全局最优位置向量 $\boldsymbol{X}_{\mathrm{opt}}$ 同时作为 FS-MMA 所有支路多模模块初始权向量的实部与虚部，即 $\boldsymbol{f}_{\mathrm{R}}^{(p)}(0) = \boldsymbol{f}_{\mathrm{I}}^{(p)}(0) = \boldsymbol{X}_{\mathrm{opt}}$，再利用式（8.3.17）与式（8.3.18）分别对 $\boldsymbol{f}_{\mathrm{R}}^{(p)}(k)$ 与 $\boldsymbol{f}_{\mathrm{I}}^{(p)}(k)$ 进行更新，以实现对各支路输入信号

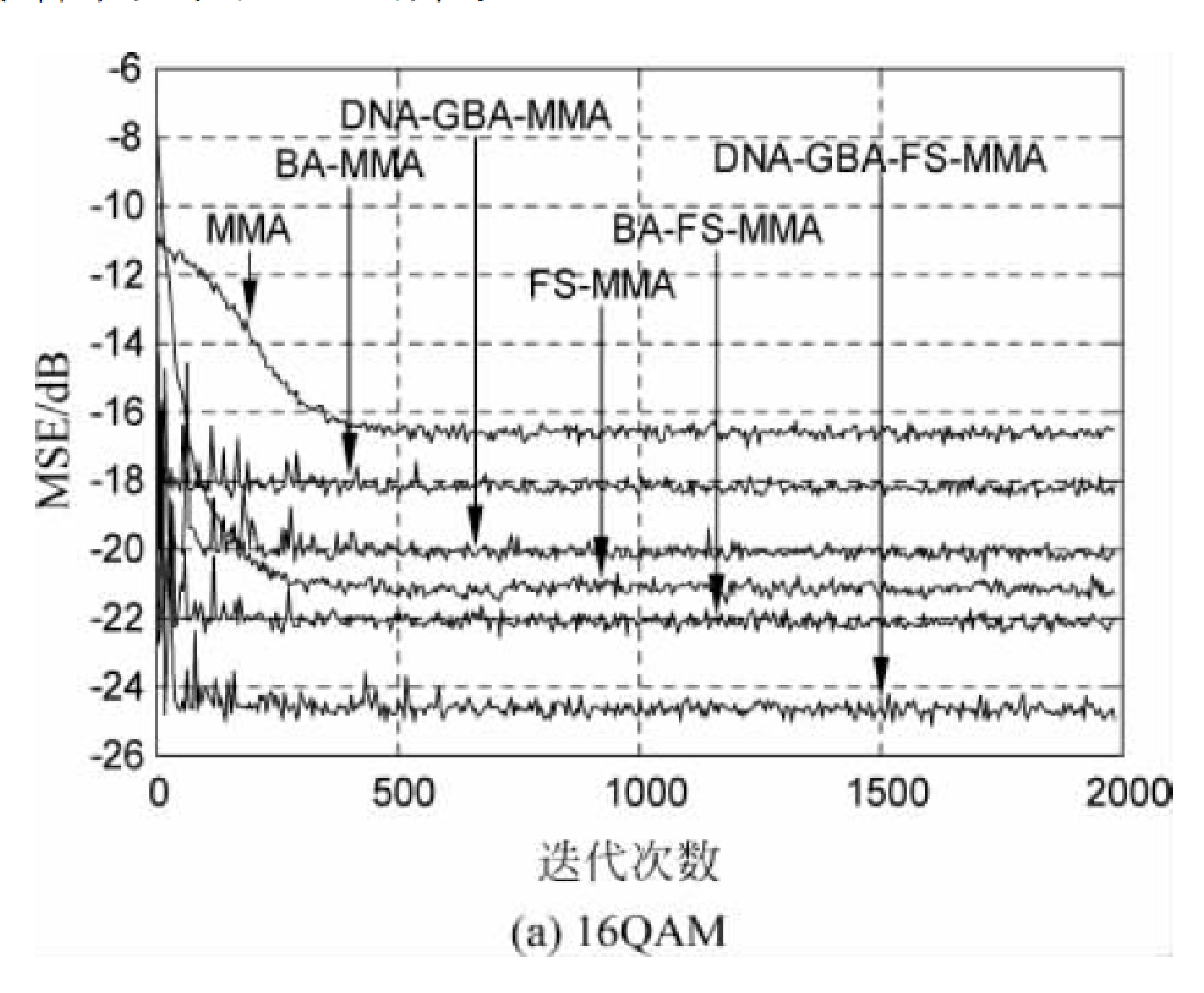
的有效均衡。各支路输出信号相加得到 DNA-GBA-FS-MMA 的输出信号 $z(k)$。

3. 算法仿真

为了验证 DNA-GBA-FS-MMA 的性能，将 MMA、FS-MMA、BA-MMA、GBA-MMA、BA-FS-MMA 与 DNA-GBA-FS-MMA 进行对比，以 T/4 分数间隔为例进行仿真实验。

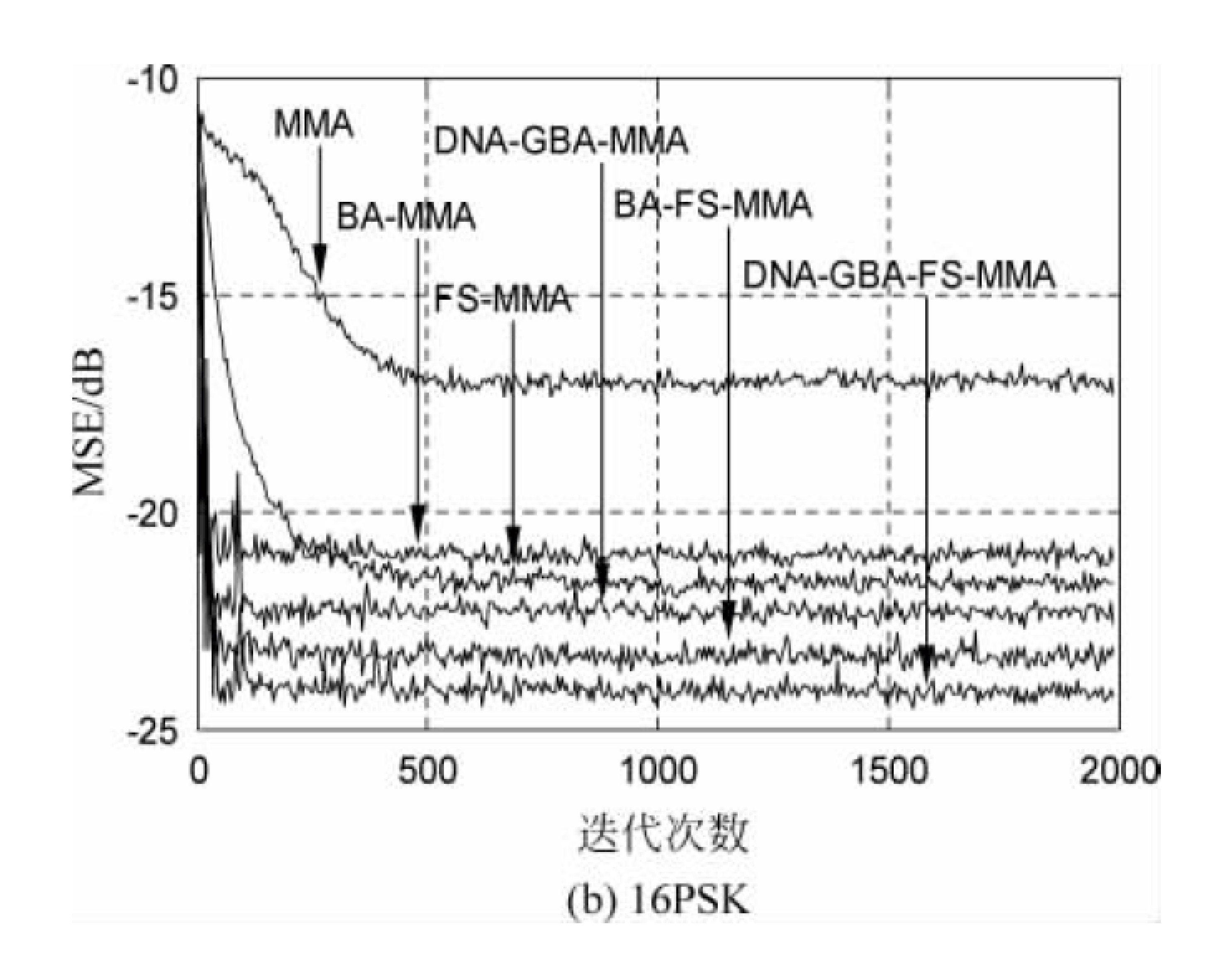
**【实验 8.5】** 参数初始化：每个蝙蝠种群数中蝙蝠个体的数量 $N=20$，频率范围[0,100]，最大响度 $A(0)=1.5$，最大频度 $r(0)=0.25$，搜索精度 $tol=1\times10^{-5}$，维数 $D=11$，响度衰减系数 $\alpha=0.9$，频度增加系数 $\gamma=0.9$，置换交叉概率 $p_z=0.8$，移位交叉概率 $p_y=0.3$，变异概率 $p_b=0.2$，最大迭代次数 $G_{max}=2000$，运行次数 runs＝2000，信道 $\boldsymbol{h}=[0.9556\quad -0.0906\quad 0.0578\quad 0.2368]$，信噪比为 25，均衡器抽头个数 $L_f=11$。

采用 16QAM 时，步长 $\mu_{MMA}=\mu_{FS\text{-}MMA}=0.02$、$\mu_{BA\text{-}MMA}=\mu_{GBA\text{-}MMA}=0.0005$、$\mu_{BA\text{-}FS\text{-}MMA}=\mu_{DNA\text{-}GBA\text{-}FS\text{-}MMA}=0.003$。采用 16PSK 时，步长 $\mu_{MMA}=\mu_{FS\text{-}MMA}=0.02$、$\mu_{BA\text{-}MMA}=\mu_{GBA\text{-}MMA}=0.0018$、$\mu_{BA\text{-}FS\text{-}MMA}=\mu_{DNA\text{-}GBA\text{-}FS\text{-}MMA}=0.0035$。仿真结果如图 8.14 所示。



(a) 16QAM

图 8.14 均方误差



(b) 16PSK

图 8.14 （续）

图 8.14(a)表明，DNA-GBA-FS-MMA 和 BA-FS-MMA 迭代 200 次左右收敛，收敛速度比 BA-MMA 和 DNA-GBA-MMA 快了约 100 步，比 FS-MMA 和 MMA 快了约 300 步；DNA-GBA-FS-MMA 的稳态误差达到约－24.5dB，比 BA-FS-MMA 降低了 2.5dB，比 FS-MMA 降低了 3.5dB，比 DNA-GBA-MMA 降低了 4.5dB，比 BA-MMA 降低了 6.5dB，比 MMA 降低了 8dB。图 8.14(b)表明，DNA-GBA-FS-MMA、BA-FS-MMA、DNA-GBA-MMA 和 BA-MMA 均迭代 100 步左右收敛，收敛速度比 FSE-MMA 和 MMA 快了约 500 步；DNA-GBA-FS-MMA 的稳态误差达到约－24dB，比 BA-FS-MMA 降低了 1dB，比 DNA-GBA-MMA 降低了 2dB，比 FS-MMA 降低了 3dB，比 BA-MMA 降低了 3.5dB，比 MMA 降低了 7.5dB。

# 参 考 文 献


[1] 李晓磊. 一种新型的智能优化算法——人工鱼群算法[D]. 杭州：浙江大学，2003.

[2] 李晓磊，钱积新. 基于分解协调的人工鱼群优化算法研究[J]. 电路与系统学报，2003，8(1)：1～6.

[3] 马宪民,刘妮.自适应视野的人工鱼群算法求解最短路径问题[J].通信学报,2014,35(1):1-6.

[4] 王培崇,雷凤君,钱旭.改进人工鱼群算法及其收敛性分析[J].科学技术与工程,2013,13(3):616-620.

[5] 王联国,洪翼.基于冯·诺依曼邻域结构的人工鱼群算法[J].控制理论与应用,2010,27(6):775-780.

[6] 张梅凤,邵诚.多峰函数优化的生境人工鱼群算法[J].控制理论与应用,2008,25(4):773-776.

[7] 俞洋,殷至锋,田亚菲.基于自适应人工鱼群算法的多用户检测器[J].电子与信息学报,2007,29(1):121-124.

[8] 翁妙凤.采用DNA进化算法优化设计TS模糊控制器[J].计算机科学,2003,30(12):141-143.

[9] 丁永生,任立红,邵世煌. DNA计算与软计算[M].北京:科学出版社,2002.

[10] 郑学东,王宾,周士华,等.基于h-距离的DNA编码序列设计[J].计算机应用,2014,34(5):1259-1262.

[11] 陈智华.基于DNA计算自组装模型的若干密码问题研究[D].武汉:华中科技大学,2009.

[12] R Deaton,R C Murphy,J A Rose,et al. A DNA based implementation of an evolutionary search for good encodings for DNA computation [C]. Evolutionary Computation,1997,IEEE International Conference on 1997,IEEE Press,1997:267-271.

[13] Weixia Yu, Ju Seok Lee, Cole Johnson, et al. Independent sets of DNA oligonucleotides for nanotechnology applications [J]. IEEE Journals & Magazines,2010,9(1):38-43.

[14] E B Baum. DNA sequences useful for computation[C]. Proceedings of the 2nd Annual DIMACS Workshop on DNA based Computers,Princeton University,1996:122-127.

[15] M Garzon, R Deaton,P Neathery,et al. On the encoding problem for DNA computing[C]. Proceedings of the 3rd DIMACS Workshop on DNA-base Computer,University of Pennsylvania,Baltimore,1997:230-237.

[16] Zacharia E, Maroulis D. An original genetic approach to the fully automatic gridding of microarray images[J]. IEEE Transactions on Medical Imaging,2008,27(6):805-813.

[17] Shang R H, Jiao L C, Gong M G, et al. Clonal selection algorithms for

dynamic multi-objective optimization[C]. Proceedings of the International Conference on Computational Inteligence and Security. Berlin: Springer-Verlag,2005,3801:846-851.

[18] Xiao Chen, Ning Wang. A DNA based genetic algorithm for parameter estimation in the hydrogenation reaction[J]. Chemical Engineering Journal, 2009,150(2-3):527-535.

[19] Zhang D, Xia Y L,He X X,et al. Multi-step evolution strategy based DNA genetic algorithm for parameters estimating [C]. The 4th International Conference on Intelligent Control and Information Processing (ICICIP). Beijing,2013:828-835.

[20] Soto Y. A method of self-recovering equalization for multilevel amplitude-modulation systems[J]. IEEE transactions on communications,1975,23(6):679-682.

[21] 郭业才. 自适应盲均衡技术[M]. 合肥:合肥工业大学出版社,2007.

[22] 郭业才. 模糊小波神经网络盲均衡理论、算法与实现[M]. 北京:科学出版社,2011.

[23] 郭业才,王丽华. 模糊神经网络控制的混合小波神经网络盲均衡算法[J]. 电子学报,2011,39(4):975-980.

[24] 郭业才. 通信信号分析与处理[M]. 合肥:合肥工业大学出版社,2009.

[25] 韩迎鸽,郭业才,吴造林,等. 基于正交小波变换的多模盲均衡器设计与算法仿真研究[J]. 仪器仪表学报,2008,29(7):1441-1445.

[26] 丁雪洁. 基于分集技术的正交小波盲均衡算法[D]. 南京:南京信息工程大学,2010.

[27] 郭业才,丁雪洁,郭福东. 正交小波分数间隔频率分集自优化盲均衡算法[J]. 兵工学报,2010,31(3):285-290.

[28] Y C Guo,H Wang,B L Zhang. DNA genetic artificial fish swarm constant modulus blind equalization algorithm and its application in medical image processing[J]. Genetics and Molecular Research,2015,14(4):11806-11813.

[29] Y C Guo, H Wang,H P Wu,et al. Multi-modulus algorithm based on global artificial fish swarm intelligent optimization of DNA encoding sequences[J]. Genetics and Molecular Research,2015,14(4):17511-17518.

[30] 王惠. DNA 人工鱼群优化盲均衡算法及 CCS 软件实现[D]. 南京:南京信息工程大学,2016.

[31] Y C Guo, H. Wang,B L Zhang. Blind equalization algorithm based on DNA

genetic optimization of artificial fish swarm[C]. International Conference on Automation,Mechanical and Electrical Engineer(AMEE). Phuket,Thailand,2015:725-732.

[32] Y C Guo, H Wang, W Huang, et al. Generalized multi-modulus blind equalization algorithm based on chaotic artificial fish swarm optimization[C]. International Conference on Automation,Mechanical and Electrical Engineer (AMEE). Phuket,Thailand,2015:774-782.

[33] H Wang, Y C Guo. A blind equalization algorithm based on global artificial fish swarm and genetic optimization DNA encoding Sequence[C]. International Industrial Informatics and Computer Engineering Conference(IIICEC). Xian,China,2015:131-134.

[34] 郭业才,王惠,吴华鹏.新变异 DNA 遗传人工鱼群优化 DNA 序列的多模算法[J].科学技术与工程,2016,16(3):66-72.

[35] Guo Yecai,Lu lu,Zhang Binglong. An Orthogonal Wavelet Transform Multi-modulus Blind Equalization Algorithm Based on Tabu Search DNA Genetic Optimization Algorithm [C]. 2015 International Conference on Materials,Engineering and Information Technology Applications(MEITA 2015),2015,Part A:129-135.

[36] 姚超然.基于 DNA 遗传蛙跳算法优化的 MIMO 盲均衡算法研究[D].南京:南京信息工程大学,2016.

[37] Zhang X, Hu X,Cui G,et al. An improved shuffled frog leaping algorithm with cognitive behavior[C]. Intelligent Control and Automation,2008. WCICA 2008. World Congress on. IEEE,2008:6197-6202.

[38] Marathe A, Condon A E,Corn R M. On combinatorial DNA word design[J]. Journal of Computational Biology A Journal of Computational Molecular Cell Biology,2001,8(3):201-219.

[39] Saranya M R, Mohan A K,Anusudha K. A composite image cipher using DNA sequence and genetic algorithm [C]. International Conference on Contemporary Computing and Informatics. IEEE,2014:1022-1026.

[40] Pan Q K, Wang L,Gao L,et al. An effective shuffled frog-leaping algorithm for lot-streaming flow shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology,2011,52(5):699-713.

[41] 崔文华,刘晓冰,王伟,等.混合蛙跳算法研究综述[J].控制与决策,2012,27(4):481-486.

[42] V Dehdeleh, A Ebrahimi, A B Nia. Improved shuffled frog leaping algorithm by using orthogonal experimental design[C]. The 2nd International Conference of Signal Processing and Intelligent Systems(ICSPIS). Tehran, Iran, 2016: 1-5.
[43] Zhou J, Dutkiewicz E, Liu R P, et al. A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems[J]. IEEE Transactions on Broadcasting, 2015, 61(4): 698-709.
[44] Ni J, Yin X, Chen J, et al. An improved shuffled frog leaping algorithm for robot path planning[C]. International Conference on Natural Computation. IEEE, 2014: 545-549.
[45] Pan Q K, Wang L, Gao L, et al. An effective shuffled frog-leaping algorithm for lot-streaming flow shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology, 2011, 52(5): 699-713.
[46] 郭业才,吴华鹏,王惠,等. 基于DNA遗传蝙蝠算法的分数间隔多模盲均衡算法[J]. 兵工学报, 2015, 36(8): 1502-1507.
[47] Huapeng Wu, Yecai Guo. DNA Genetic Optimization Bat Algorithm Based Fractionally Spaced Multi-modulus Blind Equalization Algorithm[C]. 2015 International Industrial Informatics and Computer Engineering Conference (IIICEC 2015), 2015, Part B: 581-584.
[48] Huapeng Wu, Yecai Guo. Bat Swarms Intelligent Optimization Multi-Modulus Algorithm and Influence of Modulation Mode on It[C]. 2015 International Symposium on Computers & Informatics(ISCI 2015), 2015, Part A: 83-89.
[49] Hui Wang, Yecai Guo. Novel Crossover Genetic Artificial Fish Swarm DNA Encoding Sequence Based Blind Equalization Algorithm[C]. 2015 International Symposium on Computers & Informatics(ISCI 2015), 2015, Part A: 102-109.
[50] Y C Guo, H P Wu, H Wang. Constant modulus blind equalization algorithms based on bat optimization algorithm[J]. Advanced Control, Automation and Robotics, 2015, Part 2: 546-552.
[51] 郭业才,吴华鹏. 双蝙蝠群智能优化的多模盲均衡算法[J]. 智能系统学报, 2015, 10(5): 1-7.